

# AUDIO WATERMARK RESISTANT TO MP3 COMPRESSION

---

*A THESIS SUBMITTED*

*IN PARTIAL FULFILLMENT OF REQUIREMENTS*

*FOR THE DEGREE OF*

**MASTER OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

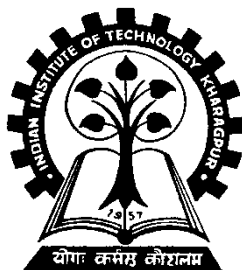
**BY**

**Muneish Adya**

**03CS3013**

**UNDER THE GUIDANCE OF**

**PROF. INDRANIL SENGUPTA**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR, INDIA**

## CERTIFICATE

This is to certify that the thesis titled “**Audio watermarks robust to MP3 compression**” is an authentic record of the original work carried out by **Mr. Muneish Adya** (Roll No. 03CS3013), under my supervision and guidance. This thesis is submitted to the Department of Computer Science and Engineering, for partial fulfillment of the requirements for the award of the degree of **Master of Technology in Computer Science and Engineering** at the **Indian Institute of Technology, Kharagpur**.

Place: I.I.T. Kharagpur

Date:

**Prof. Indranil Sengupta**

Dept. of Computer Science and Engineering

Indian Institute of Technology

Kharagpur

India 731301

## **ACKNOWLEDGEMENT**

I take great delight in expressing my deep felt gratitude to **Prof. Indranil Sengupta** under whose guidance this work was undertaken. It was an honor and pleasure to work under him. I thank him for being patient with me and taking good care to see that, I think and act in the right direction.

I express my heartfelt gratitude to my friends and also to each and every individual who was associated with this project.

Place: I.I.T. Kharagpur

Date:

**Muneish Adya**

Dept. of Computer Science and Engineering

Indian Institute of Technology

Kharagpur

India 731301

# Table of Contents

<b>Certificate .....</b>	<b>i</b>
<b>Acknowledgement .....</b>	<b>ii</b>
<b>Abstract.....</b>	<b>1</b>
<b>1 Introduction .....</b>	<b>4</b>
1.1 Digital Watermarking.....	4
1.2 Audio Watermarking.....	5
1.3 Properties of an Audio Watermark.....	6
1.3.1 Perceptual transparency .....	6
1.3.2 Watermark bit rate .....	6
1.3.3 Robustness .....	7
1.3.4 Blind or informed watermark detection.....	7
1.3.5 Security .....	7
1.3.6 Computational complexity and cost.....	8
1.4 Motivation .....	8
1.5 Organization of the thesis.....	9
<b>2 Spread Spectrum Watermark.....</b>	<b>11</b>
2.1 Common attacks on SS watermarks.....	12
2.1.1 Dual Watermarking.....	12
2.1.2 Signal processing manipulations.....	14
2.2 Performance of spread spectrum watermarks .....	15
<b>3 Human Auditory System Model.....</b>	<b>18</b>

3.1	Frequency masking .....	18
3.2	Temporal masking.....	20
<b>4</b>	<b>MP3 Compression.....</b>	<b>22</b>
4.1	Encoding.....	22
4.1.1	Layers.....	23
4.2	Decoding .....	24
<b>5</b>	<b>Spread Spectrum with Psychoacoustic Shaping .....</b>	<b>26</b>
5.1	Step I: Calculating masking thresholds.....	26
5.2	Step 2: Shaping the watermark signal .....	29
5.3	Performance of Spread Spectrum with psychoacoustic shaping scheme.....	32
<b>6</b>	<b>Discrete Wavelet Transform.....</b>	<b>35</b>
6.1	Mathematical Definition .....	36
6.1.1	One level of the transform .....	36
6.1.2	Cascading and Filter banks .....	37
6.2	Watermarking Algorithms Based on DWT.....	39
6.2.1	LSB insertion .....	39
6.2.2	Mean Quantization in DWT.....	40
<b>7</b>	<b>Spread Spectrum after DWT .....</b>	<b>42</b>
7.1	Performance of spread spectrum after DWT scheme.....	43
7.1.1	Embedding in section DD.....	43
7.1.2	Embedding in section AA.....	44
<b>8</b>	<b>Conclusions and Future Work .....</b>	<b>47</b>
8.1	Summary of Our Work.....	47
8.2	Future Scope of Work .....	47
	<b>Bibliography .....</b>	<b>48</b>

## List of Figures

<i>Figure 1.1 : an overview of the general watermarking system.....</i>	<i>5</i>
<i>Figure 2.1: Dual watermarking, pirate inserts his own watermark.....</i>	<i>13</i>
<i>Figure 2.2: Dual watermarking, pirate “subtracts off” his own watermark. ....</i>	<i>13</i>
<i>Figure 3.1: Frequency masking in the human auditory system (HAS) [1].....</i>	<i>19</i>
<i>Figure 3.2: Temporal Masking in Human Auditory System (HAS) [1].....</i>	<i>20</i>
<i>Figure 4.1: Sketch of a basic mp3 encoder.....</i>	<i>22</i>
<i>Figure 4.2: Sketch of the basic structure of the decoder. ....</i>	<i>24</i>
<i>Figure 5.1: The original host signal.....</i>	<i>26</i>
<i>Figure 5.2: Simple FFT of the original signal.....</i>	<i>27</i>
<i>Figure 5.3: Identifying tonal and non-tonal components. ....</i>	<i>27</i>
<i>Figure 5.4: Non-tonal components masked by absolute threshold are removed. ....</i>	<i>28</i>
<i>Figure 5.5: Tonal components which get masked are removed ..... </i>	<i>28</i>
<i>Figure 5.6: The complete masking prevalent in the audio. ....</i>	<i>29</i>
<i>Figure 5.7: Watermark signal to be inserted in the audio.....</i>	<i>30</i>
<i>Figure 5.8: Fast fourier transform of the watermark data.....</i>	<i>30</i>
<i>Figure 5.9: Shaping of the FFT of the watermark.....</i>	<i>31</i>
<i>Figure 5.10: Perceptually shaped watermark obtained by an inverse FFT.....</i>	<i>31</i>
<i>Figure 5.11: The final watermarked signal.....</i>	<i>32</i>
<i>Figure 6.1: Block diagram of filter analysis.....</i>	<i>37</i>
<i>Figure 6.2: A 3 level filter bank.....</i>	<i>38</i>
<i>Figure 6.3: Frequency domain representation of the DWT. ....</i>	<i>39</i>
<i>Figure 7.1: Figure showing two levels of DWT.....</i>	<i>42</i>

## List of Tables

<i>Table 2.1: SNR and PSNR values of after inserting SS watermarks .....</i>	<i>15</i>
<i>Table 2.2: Performance of SS watermarks against MP3 compression attack.....</i>	<i>16</i>
<i>Table 2.3 Performance of SS against Re-sampling attack.....</i>	<i>16</i>
<i>Table 2.4: Performance of SS against Noise addition attack.....</i>	<i>16</i>
<i>Table 5.1: Performance of SS watermark with psychoacoustic shaping against MP3 compression attack.....</i>	<i>33</i>
<i>Table 6.1: Table showing the result of cascaded DWT. ....</i>	<i>38</i>
<i>Table 7.1: SNR and PSNR values obtained after embedding .....</i>	<i>43</i>
<i>Table 7.2: Performance of spread spectrum after DWT in section DD towards MP3 attacks. ....</i>	<i>43</i>
<i>Table 7.3: Performance of spread spectrum after DWT in section DD towards re- sampling attacks.....</i>	<i>44</i>
<i>Table 7.4: Performance of spread spectrum after DWT in section DD towards white noise attacks.....</i>	<i>44</i>
<i>Table 7.5: SNR and PSNR values obtained after embedding .....</i>	<i>44</i>
<i>Table 7.6: Performance of spread spectrum after DWT in section AA towards MP3 attacks. ....</i>	<i>45</i>
<i>Table 7.7: Performance of spread spectrum after DWT in section AA towards re- sampling attacks.....</i>	<i>45</i>
<i>Table 7.8: Performance of spread spectrum after DWT in section AA towards white noise attacks.....</i>	<i>46</i>

*As the broadband communication connects the whole world together and with all the data on earth going digital, new challenges and avenues for innovation have opened up. Flexible and easy-to-use software and decreasing prices of digital devices have made it possible for people living in different parts of world to create and share multimedia data. Broadband Internet connections and near error-free transmission of data facilitate people to distribute large multimedia files and make identical digital copies of them. A possibility of perfect imitation in digital domain has made the protection of intellectual ownership and the prevention of illegal tampering of multimedia data an important technological and research issue.*

*The motivation for this work includes the provision of protection of intellectual property rights, an indication of content manipulation, and a means of protecting digital media from tampering. Digital watermarking has been proposed as a new, alternative method to enforce intellectual property rights. Digital watermarking is defined as imperceptible, robust and secure communication of ownership information through the host signal, which includes embedding into and extraction from the host signal. The main challenge in digital audio watermarking is that if the perceptual transparency parameter is fixed, the design of a watermark system cannot obtain high robustness and a high watermark data rate at the same time.*

*In the past few years a large number of algorithms for secure and robust embedding and extraction of watermarks in audio files have been developed. A broad range of embedding algorithms goes from simple Least Significant Bit (LSB) methods to various Spread Spectrum schemes. Spread Spectrum schemes have gained a lot of popularity because of their innate robustness to intelligent attacks like dual-watermarking. However, Spread Spectrum schemes fail to give a good performance against simple signal processing attacks such as mp3 compression and re-sampling. In our work, we have tried to make Spread Spectrum robust against common signal processing attacks.*



*In our first experiment to obtain it, we studied the algorithm for MP3 compression. MP3 compression uses the Human Auditory System (HAS) model in order to find out sounds that will not be heard by human ear and tries to eradicate such sounds in order to obtain imperceptible compression. It was anticipated by us that if we shape the watermark using the HAS similarly as the MP3 compression algorithm, then our watermark will become resistant to MP3 compression. The results of our experiment have been presented later in the thesis.*

*However, recently a new tool came into existence in the signal processing world. The tool is Discrete Wavelet Transform (DWT). Its main character is that it can decompose signals into different frequency components and analyzes signal in the time domain and frequency domain simultaneously. Very soon, watermarking methods based on this new transform were a fair game.*

*In our work we also studied the DWT and discovered that the transform can actually lead to an improvement in Spread Spectrum watermark. We carried out two or three levels of discrete wavelet transform and embedded water in one of the partitions using a scheme similar to spread spectrum watermark. The result was a watermarking algorithm which is robust to common signal processing attacks.*

CHAPTER 1

**INTRODUCTION**

# *1 Introduction*

---

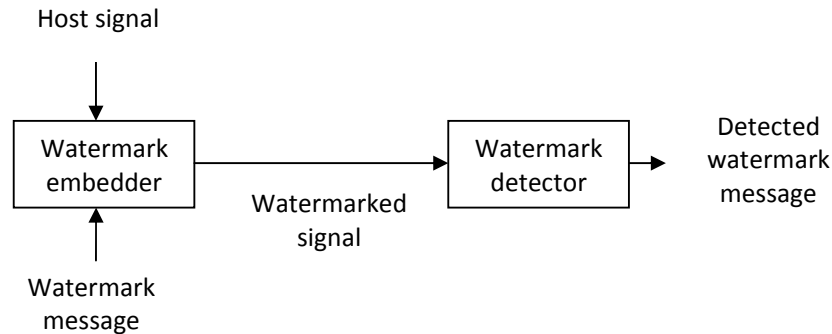
The swift development of the Internet and the digital information revolution caused major changes in the global society, ranging from the influence on the world economy to the way people nowadays communicate. Broadband communication networks and multimedia data available in a digital format (images, audio, video) opened many challenges and opportunities for innovation. Versatile and simple-to-use software and decreasing prices of digital devices (e.g. digital photo cameras, camcorders, portable CD and mp3 players, DVD players, CD and DVD recorders, laptops, PDAs) have made it possible for consumers from all over the world to create, edit and exchange multimedia data. Broadband Internet connections and almost an errorless transmission of data facilitate people to distribute large multimedia files and make identical digital copies of them.

Digital media files do not suffer from any quality loss due to multiple copying processes, such as analogue audio and VHS tapes. Furthermore, recording medium and distribution networks for analogue multimedia are more expensive. These first-view advantages of digital media over the analogue ones transform to disadvantages with respect to the intellectual rights management because a possibility for unlimited copying without a loss of fidelity cause a considerable financial loss for copyright holders. The ease of content modification and a perfect reproduction in digital domain have promoted the protection of intellectual ownership and the prevention of the unauthorized tampering of multimedia data to become an important technological and research issue.

## **1.1 Digital Watermarking**

Digital watermarking has been proposed as a new, alternative method to enforce the intellectual property rights and protect digital media from tampering. It involves a process of embedding into a host signal a perceptually transparent digital signature, carrying a message about the host signal in order to "mark" its ownership. The digital signature is called the digital watermark. The digital watermark contains data that can be used in various applications, including digital rights management, broadcast

monitoring and tamper proofing. Although perceptually transparent, the existence of the watermark is indicated when watermarked media is passed through an appropriate watermark detector. Figure 1.1 shows a block diagram of a simple watermarking system.



**Figure 1.1 : an overview of the general watermarking system**

A watermark, which usually consists of a binary data sequence, is inserted into the host signal in the watermark embedder. Thus, a watermark embedder has two inputs, one is the watermark message (usually accompanied by a secret key) and the other is the host signal (e.g. image, video clip, audio sequence etc.). The output of the watermark embedder is the watermarked signal, which cannot be perceptually discriminated from the host signal.

## 1.2 Audio Watermarking

Digital audio watermarking involves the hiding of data within a digital audio file. Applications for this technology are numerous. Intellectual property protection is currently the main driving force behind research in this area. To combat online music piracy, a digital watermark could be added to all recording prior to release, signifying not only the author of the work, but the user who has purchased a legitimate copy. Newer operating systems equipped with digital rights management (DRM) software will extract the watermark from audio files prior to playing them on the system. The DRM software will ensure that the user has paid for the song by comparing the watermark to the existing purchased licenses on the system.

Other non-rights related uses for watermarking technology include embedding auxiliary information which is related to a particular song, like lyrics, album information, or a small web page, etc. Watermarking could be used in voice conferencing systems to indicate to others which party is currently speaking. A video application of this technology would consist of embedding subtitles or closed captioning information as a watermark.

## **1.3 Properties of an Audio Watermark**

A watermarking algorithm can be characterized by a number of properties. The relative importance of each property however depends on the demands of the application. The six important properties are as follows.

### **1.3.1 Perceptual transparency**

In all most every application, the watermark-embedding algorithm has to insert watermark data without changing the perceptual quality of the host audio signal. The fidelity of a watermarking algorithm is usually defined as a perceptual similarity between the original and watermarked audio sequence. However, the quality of the watermarked audio may get tainted, either intentionally by an adversary or unintentionally during the transmission process, before a person perceives it. In such a case, it is more sensible to redefine the fidelity of a watermarking algorithm as a perceptual similarity between the watermarked audio and the original host audio at the point at which they are presented to a consumer.

### **1.3.2 Watermark bit rate**

Bit rate of an embedded watermark is defined as the number of bits of the watermark embedded in one second of the host audio signal and is given in bits per second (bps). The bps requirement of a watermark depends on the application. For example, some applications, such as copy control, require the insertion of a serial number or author ID, with the average bit rate of 0.5 bps. In some envisioned applications, like hiding speech in audio, algorithms have to be able to embed watermarks with the bit rate that is a significant fraction of the host audio bit rate, i.e. up to 150 kbps.

### **1.3.3 Robustness**

The robustness of a watermarking algorithm is defined as its ability to detect/ extract the watermark after common signal processing manipulations. The set of signal processing modifications to which a watermarking algorithm needs to be robust against is completely application dependent. For example, in radio broadcast monitoring, embedded watermark need only to survive distortions caused by the transmission process, including dynamic compression and low pass filtering, because the watermark detection is done directly from the broadcast signal. On the other hand, in some algorithms robustness is completely undesirable and those algorithms are labeled fragile audio watermarking algorithms.

### **1.3.4 Blind or informed watermark detection**

In some applications, a detection algorithm may use the original host audio to extract watermark from the watermarked audio sequence (informed detection). It often significantly improves the detector performance, in that the original audio can be subtracted from the watermarked copy, resulting in the watermark sequence alone. However, if detection algorithm does not have access to the original audio (blind detection) and this inability substantially decreases the amount of data that can be hidden in the host signal. The complete process of embedding and extracting of the watermark is modeled as a communications channel where watermark is distorted due to the presence of strong interference and channel effects [33]. A strong interference is caused by the presence of the host audio, and channel effects correspond to signal processing operations.

### **1.3.5 Security**

Watermark algorithm must be secure in the sense that an adversary must not be able to detect the presence of embedded data, let alone remove the embedded data. The security of watermark process is interpreted in the same way as the security of encryption techniques and it cannot be broken unless the authorized user has access to a secret key that controls watermark embedding. An unauthorized user should be unable to extract the data in a reasonable amount of time even if he knows that the host signal contains a watermark and is familiar with the exact watermark embedding algorithm. Security requirements vary with application and the most stringent are in

cover communications applications, and, in some cases, data is encrypted prior to embedding into host audio.

### **1.3.6 Computational complexity and cost**

The implementation of an audio watermarking system is a tedious task, and it depends on the business application involved. The principal issue from the technical point of view is the computational complexity of embedding and detection algorithms and the number of embedders and detectors used in the system. For example, in broadcast monitoring, embedding and detection must be done in real time, while in copyright protection applications, time is not a crucial factor for a practical implementation. One of the economic issues is the design of embedders and detectors, which can be implemented as hardware or software plug-ins, is the difference in processing power of different devices (laptop, PDA, mobile phone, etc.).

## **1.4 Motivation**

A fair use of multimedia data combined with a fast delivery of multimedia to users having different devices with a fixed quality of service is becoming a challenging and important topic. Traditional methods for copyright protection of multimedia data are no longer sufficient. Hardware-based copy protection systems have already been easily circumvented for analogue media. Hacking of digital media systems is even easier due to the availability of general multimedia processing platforms, e.g. a personal computer. Simple protection mechanisms that were based on the information embedded into header bits of the digital file are useless because header information can easily be removed by a simple change of data format, which does not affect the fidelity of media.

In the past few years a large number of algorithms for secure and robust embedding and extraction of watermarks in audio files have been developed. A broad range of embedding algorithms goes from simple Least Significant Bit (LSB) methods to various Spread Spectrum schemes. Spread Spectrum schemes have gained a lot of popularity because of their innate robustness to intelligent attacks like dual-watermarking. However, Spread Spectrum schemes fail to give a good performance against simple signal processing attacks such as mp3 compression and re-sampling.

The motivation of our work lies in making spread spectrum technology robust to even simple signal manipulations so that audio media files of the world can enjoy robust security.

## **1.5 Organization of the thesis**

A summary of the contents of the chapters to follow is given below.

Chapter 2 is an introduction to spread spectrum watermarking. The scheme was implemented by us and the results have been displayed.

Chapter 3 talks about the human Auditory System (HAS) model.

Chapter 4 discusses the MP3 compression.

In chapter 5 we discuss an algorithm to shape the watermark before embedding to make it robust to MP3 compression. The results have also been displayed in the same chapter.

Chapter 6 discusses a new tool in signal processing i.e. discrete wavelet transform.

Chapter 7 discusses our algorithm to insert spread spectrum watermark after discrete wavelet transform.

Chapter 8 talks about conclusion and future work.



CHAPTER 2

**SPREAD SPECTRUM**

**WATERMARK**

## 2 Spread Spectrum Watermark

---

In a normal communication channel, it is often desirable to concentrate the information in as narrow a region of the frequency spectrum as possible in order to conserve available bandwidth and to reduce power. The basic spread spectrum technique, on the other hand, is designed to encode a stream of information by spreading the encoded data across as much of the frequency spectrum as possible. This allows the signal reception, even if there is interference on some frequencies. While there are many variations on spread spectrum communication, we concentrated on Direct Sequence Spread Spectrum encoding (DSSS).

Let us denote as  $x$  the original signal vector to be watermarked. It represents a block of samples of the original audio signal. The corresponding watermarked vector is generated simply by:

$$y = x + w \quad \dots\dots\dots (2.1)$$

, where the watermark  $w$  is has elements  $w(i)$  that can assume one of two equiprobable values, i.e.  $w(i) \in \{-\Delta, +\Delta\}$ , independently of  $x$ . Parameter  $\Delta$  should be set based on the sensitivity of the HAS to amplitude changes. In our case,  $x$  is a vector of magnitude frequency components in a decibel scale, so  $\Delta$  should not be higher than about 1 dB. A correlation detector performs the optimal test for the presence of the watermark

$$C = y \cdot w = (x + w) \cdot w = x \cdot w + N \cdot \Delta^2 \quad \dots\dots\dots (2.2)$$

, where  $N$  is the cardinality of the vectors. Since the original audio file is completely uncorrelated to the vector  $w$  the product ' $x \cdot w$ ' theoretically should turns out to be nearly zero. The optimal detection rule is to declare the watermark present if  $C > T$ . The choice of the threshold  $T$  controls the tradeoff between false alarm and detection probabilities.

In our experiments, however, the watermarking method used was a little different from DSSS and is described below.

Firstly, the whole audio file is divided into partitions and one bit is hidden in one partition/block as follows.

Vector  $x$  is considered to be a block of the original host signal. A secret key  $K$  is used by a pseudo random number generator (PRN) to produce a chip sequence with zero mean and whose elements are equal to  $+\Delta$  or  $-\Delta$ . Let this be denoted as sequence  $u$ . The sequence  $u$  is then added to or subtracted from the signal  $x$  according to the variable  $b$ , the data bit to be hidden in this block, where  $b$  assumes the values 1 or 0.

Hence embedding can be performed as:

$$y = x + (2b-1) * u \quad \dots\dots\dots (2.3)$$

, where  $b$  takes value 1 or 0 ( i.e.  $2b-1$  takes value +1 or -1 accordingly). During watermark extraction, for each block  $y$  of the watermarked audio,  $y.u$  is calculated.

$$b' = y.u = (x + (2b-1)u).u = x.u + (2b-1). N.\Delta^2 \quad \dots\dots\dots (2.4)$$

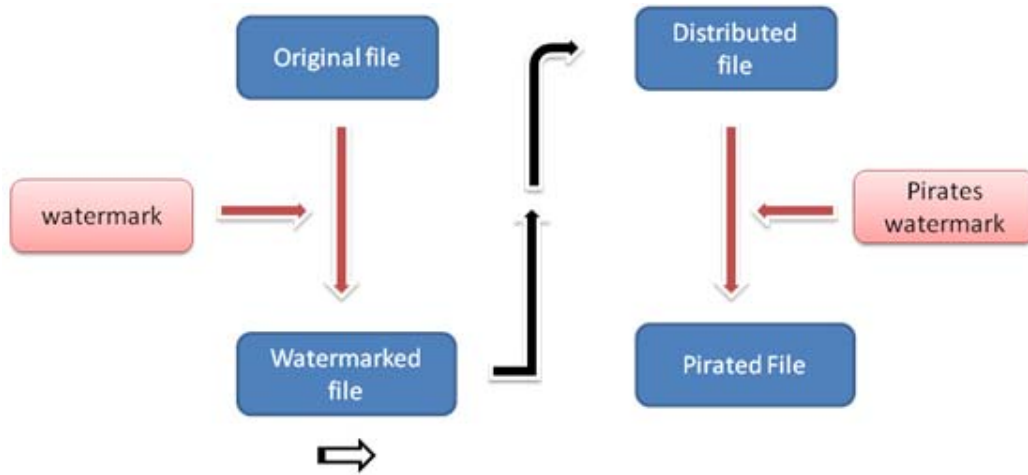
Since  $x.u$  is nearly zero ( $x$  and  $u$  being uncorrelated) hence if  $b'$  is positive, data bit hidden in that block is taken to be 1 else if the value is negative data bit hidden is taken to be 0.

## 2.1 Common attacks on SS watermarks

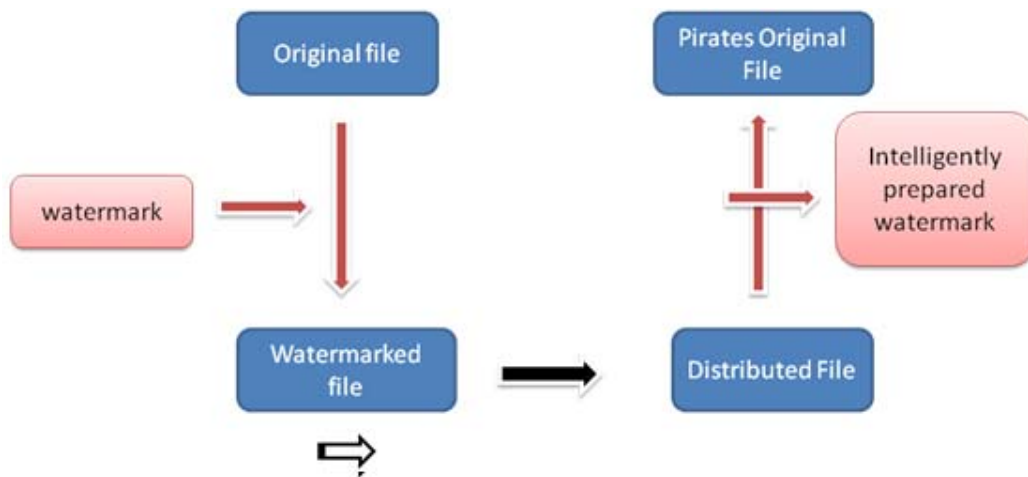
As and when a successful watermarking scheme is developed, a number of attacks to destroy the watermarking algorithm also crop up simultaneously. The following few pages describe some of the intelligent and simple signal manipulation attacks and the methods developed (if possible) to counter the attacks.

### 2.1.1 Dual Watermarking

The main aim of a watermarking method is to establish and protect the rightful ownership of audio file unambiguously. However many watermarking schemes are unable to resolve ownership when multiple claims on an audio file are made. The problem of dual watermarking can be simply explained by the following flow diagrams.



**Figure 2.1: Dual watermarking, pirate inserts his own watermark**



**Figure 2.2: Dual watermarking, pirate “subtracts off” his own watermark.**

Watermarking methods that do not require the original host signal for detection / extraction are the ones that are most susceptible to the problem of dual watermarking. A pirate simply adds his or her watermark to the watermarked data. The data now has two watermarks. Currently, many watermarking schemes are unable to establish who watermarked the data first.

Watermarking procedures that require the original data set for watermark detection also suffer from deadlocks. In such schemes, a party other than the owner may

counterfeit a watermark by “subtracting off” a second watermark (as shown in figure 2) from the publicly available data and claim the result to be his or her original. This second watermark allows the pirate to claim copyright ownership since he or she can show that both the publicly available data and the original of the rightful owner contain a copy of their counterfeit watermark.

The problem of dual watermark had been described as an unsolvable problem but [ppr on dual] gives a very elegant and simple solution which can be applied to spread spectrum to make it robust to this attack. The author specifies the use of two seeds  $x^1$  and  $x^2$  to seed the pseudorandom sequence generator from which the noisy-pseudorandom sequence  $\mathbf{u}$  can be generated. Only when the two keys are present, the watermark can be extracted. Without the availability of two keys, the watermark can neither be extracted nor removed. The noise like sequence  $\mathbf{u}$  is the actual watermark that is embedded into the host signal. The key  $x^1$  is owner dependent i.e. the secret key assigned or given by the original owner. The key  $x^2$  is signal dependent i.e. it is computed from the host audio signal which the owner wishes to watermark.

The signal dependent key  $x^2$  makes it extremely difficult for a pirate to introduce or subtract off his counterfeit watermark. The pirate can only provide the key  $x^1$  to the arbitrator. The key  $x^2$  is produced by the watermarking algorithm from the original signal itself. Hence for the pirate to produce successful piracy, the pirate must generate a watermark which creates a counterfeit original that can generate the watermark again. Such a procedure is computationally infeasible and hence in this way, spread spectrum watermark can be made robust to dual watermarking attack.

## **2.1.2 Signal processing manipulations**

There are a large number of simple signal processing attacks, which completely fails the spread spectrum watermarking scheme. Some of them are:

### **2.1.2.1 MP3 compression:**

In this attack, the watermarked file is compressed using layer I mp3 compression model with different compression schemes such as 128 kbps or 64 kbps. Then the mp3 file is de-compressed and converted back to .wav file.

Spread spectrum watermarking does not stand robust to such attacks and loses all its information.

### **2.1.2.2 Re-sampling attack:**

This is also a very simple yet damaging attack. The watermarked file is taken and re-sampled with a different frequency, usually K times smaller than the original sampling frequency. Then the resulting audio file is re-sampled back to the original frequency. Currently, spread spectrum watermarking stays vulnerable to re-sampling attacks when value of K becomes greater than 3.













## **2.2 Performance of spread spectrum watermarks**

The performance of this scheme against common signal processing manipulations is shown in the following tables. The definitions of SNR, PSNR and NC are given in the appendix.

























**Table 2.1: SNR and PSNR values of after inserting SS watermarks**

<b>File name</b>	<b>SNR</b>	<b>PSNR</b>
<b>Piano.wav</b>	24.2843	40.0003
<b>Orchestra.wav</b>	18.0983	40.0003
<b>Beats.wav</b>	20.3044	40.2859
<b>Song.wav</b>	18.7847	40.0003
<b>Pop.wav</b>	16.0951	42.0662
<b>Jazz.wav</b>	25.5969	40.0003







**Table 2.2: Performance of SS watermarks against MP3 compression attack**

Attack Category: MP3 attack			Watermarking scheme: Spread Spectrum			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
MP3 compression, 128kbps	 NC = 0.4724	 NC = 0.4625	 NC = 0.4746	 NC = 0.4338	 NC = 0.4879	 NC = 0.4614
MP3 compression, 64kbps	 NC = 0.5055	 NC = 0.4967	 NC = 0.4956	 NC = 0.5055	 NC = 0.4724	 NC = 0.4945

**Table 2.3 Performance of SS against Re-sampling attack.**

Attack Category: Re-sampling attack			Watermarking scheme: Spread Spectrum			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Re-sampling factor = 2	 NC = 0.8940	 NC = 0.9757	 NC = 0.9503	 NC = 0.9735	 NC = 0.9956	 NC = 0.8642
Re-sampling factor = 3	 NC = 0.8168	 NC = 0.9382	 NC = 0.9492	 NC = 0.8764	 NC = 0.9801	 NC = 0.7759
Re-sampling factor = 4	 NC = 0.6987	 NC = 0.8477	 NC = 0.8146	 NC = 0.8444	 NC = 0.9272	 NC = 0.7185
Re-sampling factor = 5	 NC = 0.6402	 NC = 0.7903	 NC = 0.7660	 NC = 0.8091	 NC = 0.8885	 NC = 0.6898

**Table 2.4: Performance of SS against Noise addition attack**

Attack Category: White Noise addition			Watermarking scheme: Spread Spectrum			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Gaussian Noise	 NC = 0.7461	 NC = 0.7439	 NC = 0.8256	 NC = 0.7450	 NC = 0.7075	 NC = 0.7208

CHAPTER 3

**THE HUMAN AUDITORY**

**SYSTEM MODEL**



## *3 Human Auditory System Model*

---

The HAS perceives sounds over a range of power greater than 109:1 and a range of frequencies greater than 103:1. The sensitivity of the HAS to the additive white Gaussian noise (AWGN) is high as well; this noise in a sound file can be detected as low as 70 dB below ambient level.

On the other hand, opposite to its large dynamic range, HAS contains a fairly small differential range, i.e. loud sounds generally tend to mask out weaker sounds. Additionally, HAS is insensitive to a constant relative phase shift in a stationary audio signal and some spectral distortions interprets as natural, perceptually non-annoying ones.

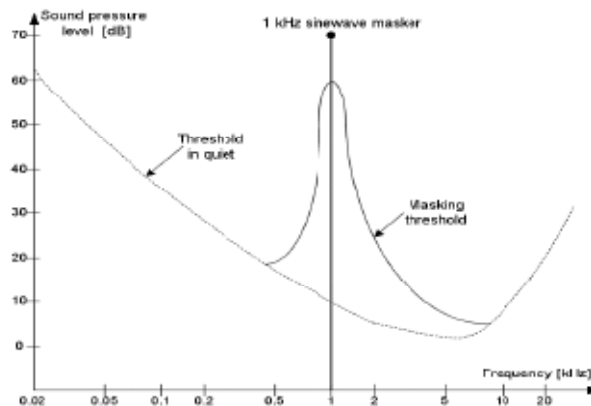
Auditory perception is based on the critical band analysis in the inner ear where a frequency-to-location transformation takes place along the basilar membrane. The power spectra of the received sounds are not represented on a linear frequency scale but on limited frequency bands called critical bands. The auditory system is usually modeled as a band pass filter-bank, consisting of strongly overlapping band pass filters with bandwidths around 100 Hz for bands with a central frequency below 500 Hz and up to 5000 Hz for bands placed at high frequencies. If the highest frequency is limited to 24000 Hz, 26 critical bands have to be taken into account.

Two properties of the HAS dominantly used in watermarking algorithms are frequency(simultaneous) masking and temporal masking.

### **3.1 Frequency masking**

Frequency (simultaneous) masking is a frequency domain phenomenon where a low level signal, e.g. a pure tone (the maskee), can be made inaudible (masked) by a simultaneously appearing stronger signal (the masker), e.g. a narrow band noise, if the masker and maskee are close enough to each other in frequency. A masking threshold can be derived below which any signal will not be audible. The masking threshold depends on the masker and on the characteristics of the masker and maskee

(narrowband noise or pure tone). For example, with the masking threshold for the sound pressure level (SPL) equal to 60 dB, the masker in figure 3.1 at around 1 kHz, the SPL of the maskee can be surprisingly high - it will be masked as long as its SPL is below the masking threshold. The slope of the masking threshold is steeper toward lower frequencies; in other words, higher frequencies tend to be more easily masked than lower frequencies. It should be pointed out that the distance between masking level and masking threshold is smaller in noise-masks tone experiments than in tone-masks-noise experiments due to HAS's sensitivity toward additive noise. Noise and low-level signal components are masked inside and outside the particular critical band if their SPL is below the masking threshold. Noise contributions can be coding noise, inserted watermark sequence, aliasing distortions, etc. Without a masker, a signal is inaudible if its SPL is below the threshold in quiet, which depends on frequency and covers a dynamic range of more than 70 dB as depicted in the lower curve of Figure 3.1.



**Figure 3.1: Frequency masking in the human auditory system (HAS) [1].**

The distance between the level of the masker and the masking threshold is called signal-to-mask ratio. Its maximum value is at the left border of the critical band. Within a critical band, noise caused by watermark embedding will be audible as long as signal-to-noise ratio (SNR) for the critical band is higher than its SMR. Let  $SNR(m)$  be the signal-to-noise ratio resulting from watermark insertion in the critical

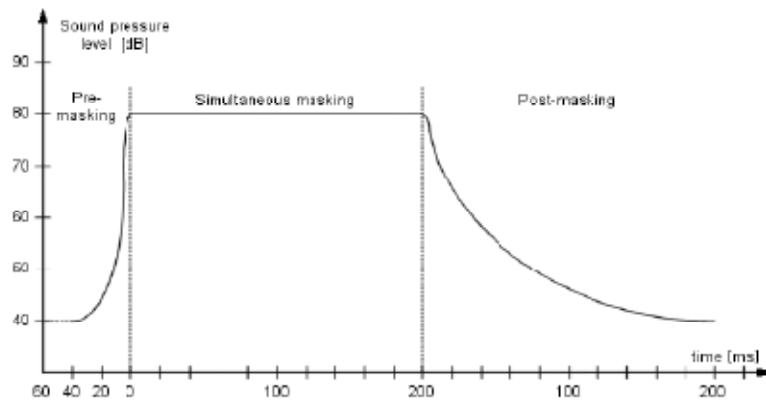
band  $m$ ; the perceivable distortion in a given sub-band is then measured by the noise to mask ratio:

$$\text{NMR}(m) = \text{SMR} - \text{SNR}(m) \quad \dots\dots\dots (2.2)$$

The noise-to-mask ratio  $\text{NMR}(m)$  expresses the difference between the watermark noise in a given critical band and the level where a distortion may just become audible; its value in dB should be negative.

### 3.2 Temporal masking

In addition to frequency masking, two phenomena of the HAS in the time domain also play an important role in human auditory perception. Those are pre-masking and post-masking in time. The temporal masking effects appear before and after a masking signal has been switched on and off, respectively (Figure 3.2). The duration of the pre-masking is significantly less than one-tenth that of the post-masking, which is in the interval of 50 to 200 milliseconds. Both pre- and post-masking have been exploited in the MPEG audio compression algorithm and several audio watermarking methods.



**Figure 3.2: Temporal Masking in Human Auditory System (HAS) [1].**

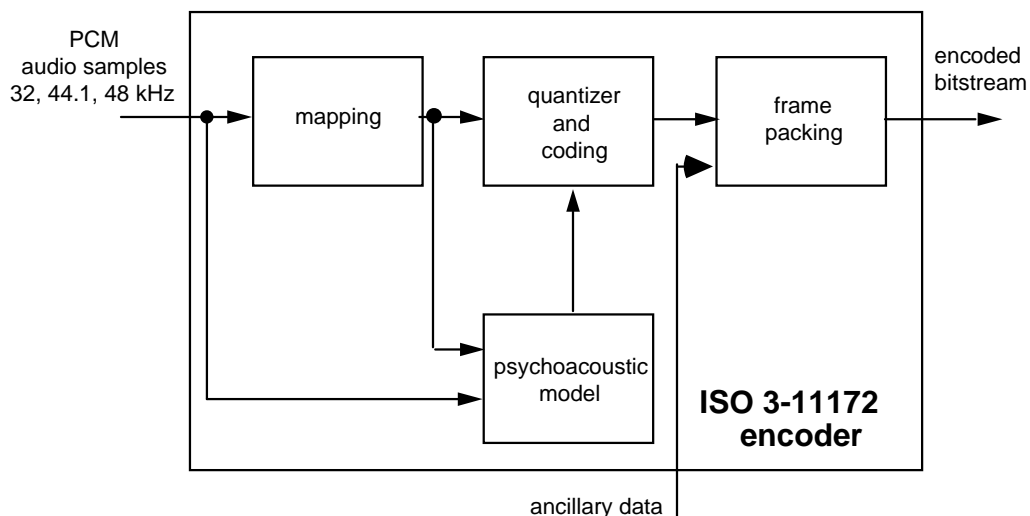
CHAPTER 4

**MP3 COMPRESSION**

MPEG was formed in 1988 to establish a standard for the coded representation of moving pictures and associated audio stored on digital storage media.

### 4.1 Encoding

The encoder processes the digital audio signal and produces the compressed bit stream for storage. The encoder algorithm is not standardized, and may use various means for encoding such as estimation of the auditory masking threshold, quantization, and scaling.



**Figure 4.1: Sketch of a basic mp3 encoder.**

Input audio samples are fed into the encoder. The mapping creates a filtered and sub sampled representation of the input audio stream. The mapped samples may be called either sub-band samples (as in Layer I or II, see below) or transformed sub-band samples (as in Layer III). A psychoacoustic model creates a set of data to control the quantizer and coding. These data are different depending on the actual coder

implementation. One possibility is to use an estimation of the masking threshold to do this quantizer control. The quantizer and coding block creates a set of coding symbols from the mapped input samples. Again, this block can depend on the encoding system. The block 'frame packing' assembles the actual bitstream from the output data of the other blocks, and adds other information (e.g. error correction) if necessary.

There are four different modes possible, single channel, dual channel (two independent audio signals coded within one bitstream), stereo (left and right signals of a stereo pair coded within one bitstream), and joint stereo (left and right signals of a stereo pair coded within one bitstream with the stereo irrelevancy and redundancy exploited).

### **4.1.1 Layers**

Depending on the application, different layers of the coding system with increasing encoder complexity and performance can be used. An ISO/MPEG Audio Layer N decoder is able to decode bitstream data which has been encoded in Layer N and all layers below N.

#### ***4.1.1.1 Layer I:***

This layer contains the basic mapping of the digital audio input into 32 subbands, fixed segmentation to format the data into blocks, a psychoacoustic model to determine the adaptive bit allocation, and quantization using block companding and formatting. The theoretical minimum encoding/decoding delay for Layer I is about 19 ms.

#### ***4.1.1.2 Layer II:***

This layer provides additional coding of bit allocation, scalefactors and samples. Different framing is used. The theoretical minimum encoding/decoding delay for Layer II is about 35 ms.

#### ***4.1.1.3 Layer III:***

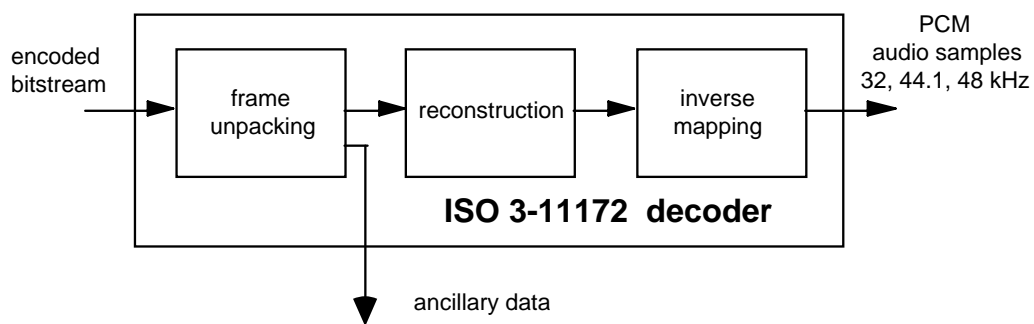
This layer introduces increased frequency resolution based on a hybrid filter-bank. It adds a different (non-uniform) quantizer, adaptive segmentation and entropy coding

of the quantized values. The theoretical minimum encoding/decoding delay for Layer III is about 19 ms.

Joint stereo coding can be added as an additional feature to any of the layers.

## 4.2 Decoding

The decoder accepts the compressed audio bit-stream, decodes the data elements, and uses the information to produce digital audio output.



**Figure 4.2: Sketch of the basic structure of the decoder.**

Bit-stream data is fed into the decoder. The bit-stream unpacking and decoding block does error detection if error-check is applied in the encoder. The bit-stream data are unpacked to recover the various pieces of information. The reconstruction block reconstructs the quantized version of the set of mapped samples. The inverse mapping transforms these mapped samples back into uniform PCM.

CHAPTER 5

**SPREAD SPECTRUM WATERMARK  
WITH PSYCHOACOUSTIC SHAPING**



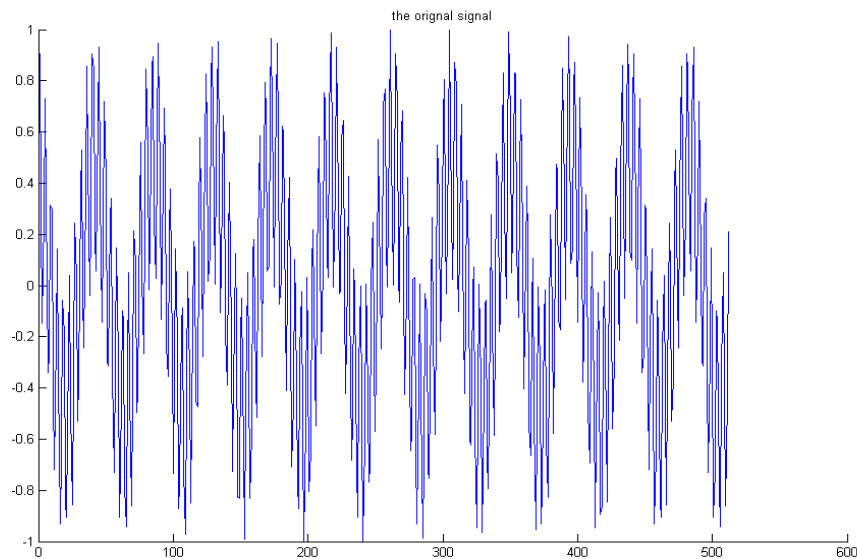
## 5 Spread Spectrum with Psychoacoustic Shaping

---

For this experiment, the MP3 compression-type algorithm was used to come up with a shaping of the sequence  $u$  before it is embedded into the original audio file. It was expected that such a watermark will be resistant to MP3 attack. The steps followed in this watermarking method have been explained below using images (which were generated during the embedding process). The audio file used simply consists of two sinusoidal waves. The results are in end.

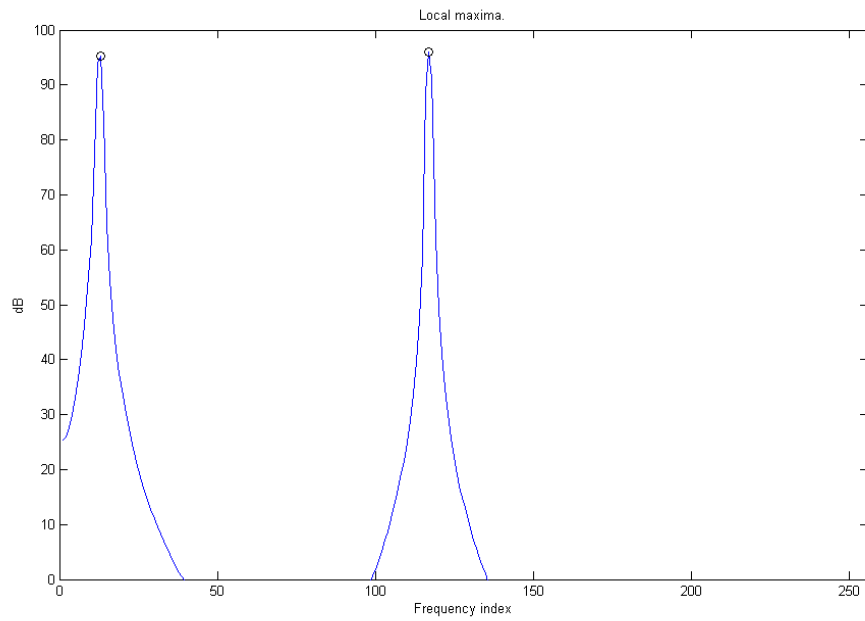
### 5.1 Step I: Calculating masking thresholds

In this step we try to obtain the masking thresholds for a block of the audio file using an algorithm similar to the MP3 compression model layer I. The following pages describe the complete process using images and in order to make the process easy to understand. Figure 5.1 shows the original audio signal to be watermarked, which is a simple superposition of two sinusoidal waves.



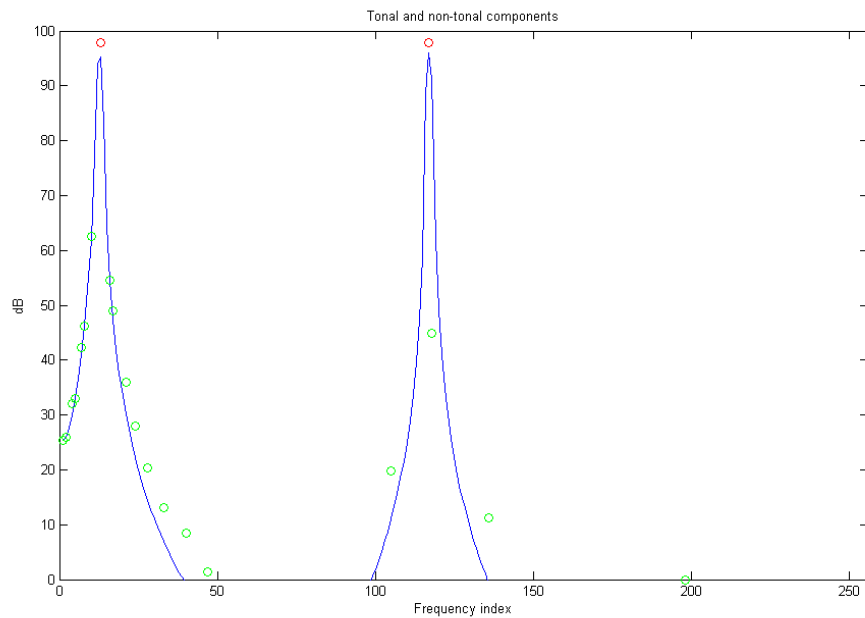
**Figure 5.1: The original host signal**

Figure 5.2 shows the fast fourier transform of the host signal. It can be clearly seen that FFT results in two peaks representing the two sinusoidal waves of the host signal.



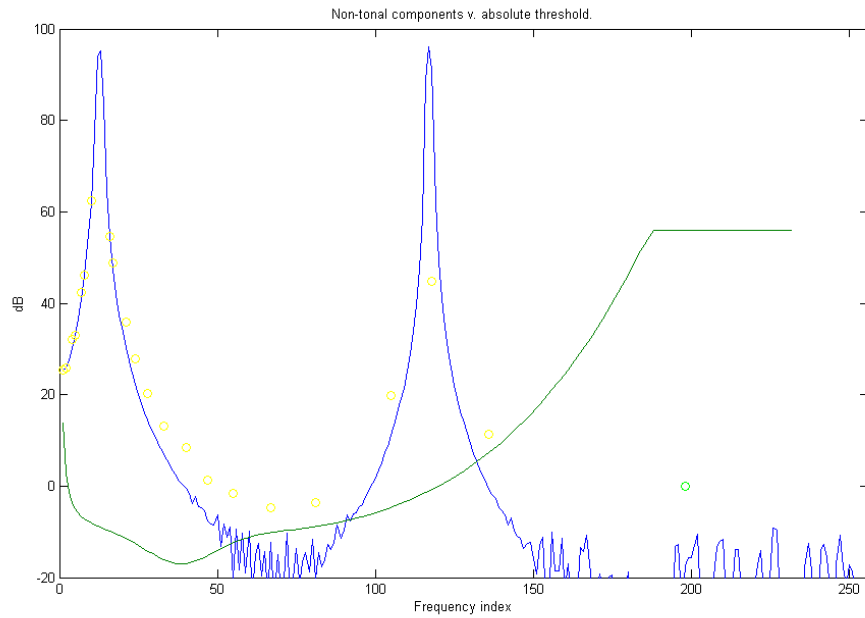
**Figure 5.2: Simple FFT of the original signal**

In figure 5.3, the tonal and non-tonal components have been identified and marked with red and green points respectively. Components that are at a peak are called the tonal components and they lend the maximum sound to the audio.



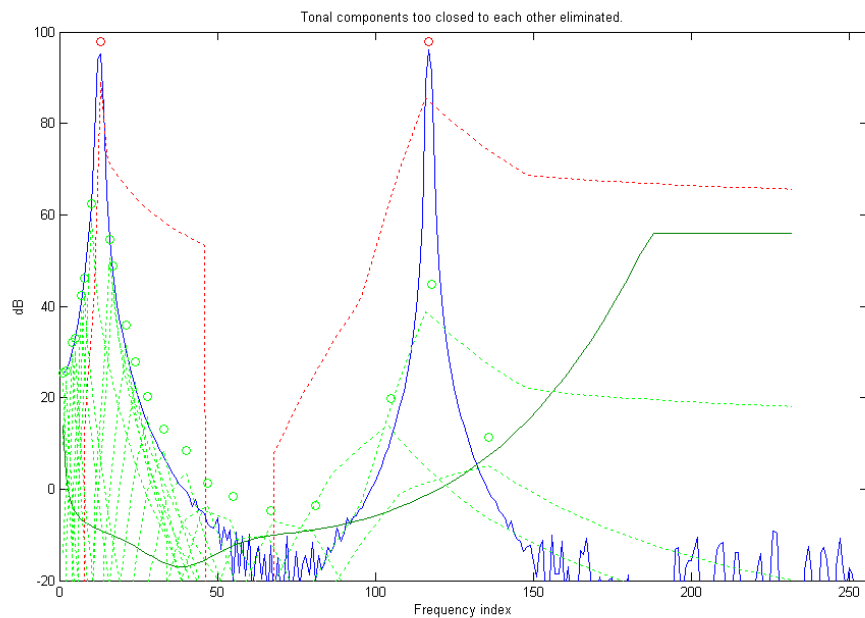
**Figure 5.3: Identifying tonal and non-tonal components.**

Figure 5.4 shows the curve of absolute threshold (threshold in silence) in green. Components with power smaller than absolute threshold are removed. According to frequency masking, tonal components too close mask each other.



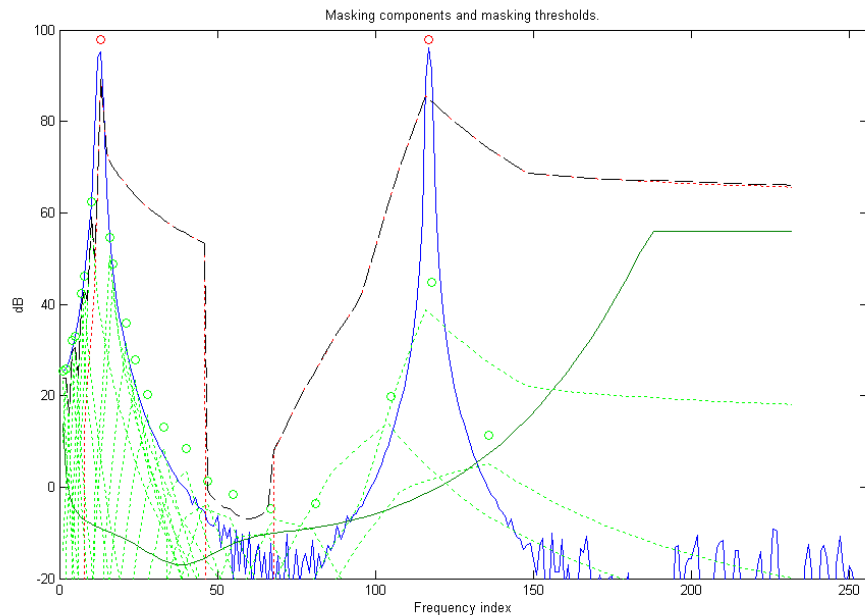
**Figure 5.4: Non-tonal components masked by absolute threshold are removed.**

Figure 5.5 shows that masked tonal components are removed since they will be inaudible to human ear. It also shows the masking effect produced by tonal components (in red) and non-tonal components (in green).



**Figure 5.5: Tonal components which get masked are removed**

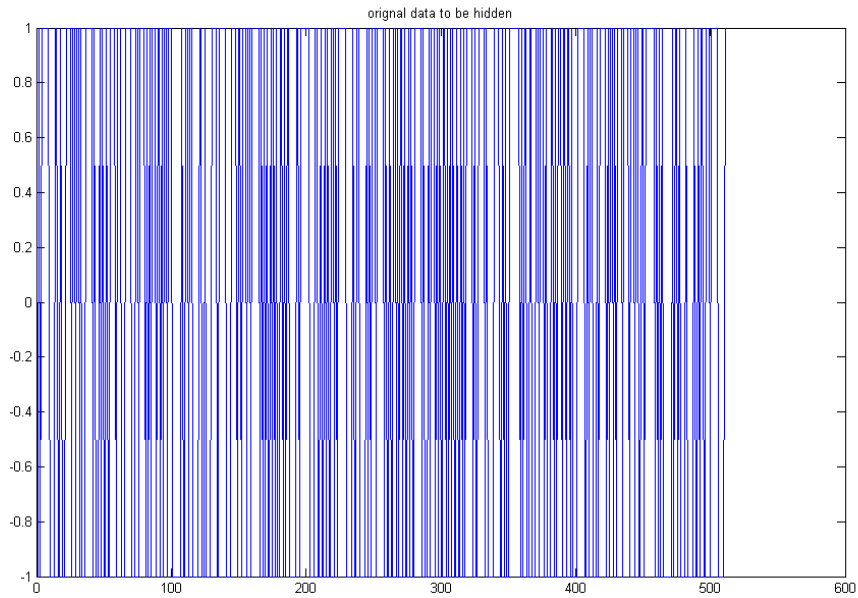
Figure 5.6 shows the masking effect in the audio file. At any point, masking can be due to tonal components, non-tonal components or the absolute threshold, whichever is maximum.



**Figure 5.6: The complete masking prevalent in the audio.**

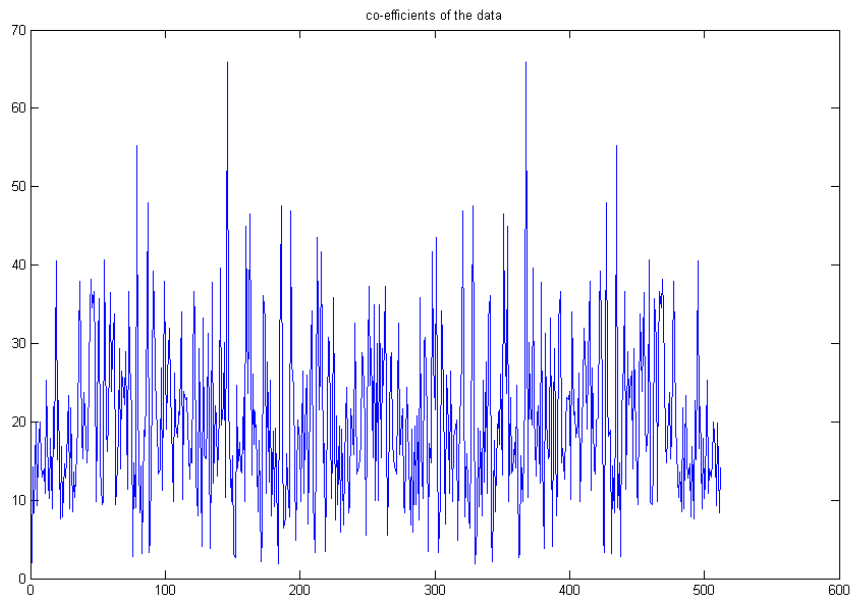
## 5.2 Step 2: Shaping the watermark signal

The masking thresholds that were obtained in the previous step are used to shape the watermark signal before it is inserted into the original audio. The idea was that if a watermark signal had been shaped then it will become resistant to MP3 compression, which uses a similar masking algorithm for its compression process. A lot of experiments were conducted in order to find out how the shaping must be done in order to obtain the best performance. Finally a simple multiplication of the masking threshold values and the FFT of the watermark to be inserted was used as the shaping algorithm. The following few pages describe the process with the help of images making the concept clearer. Figure 5.7 shows the noise like sequence which needs to be hidden inside the host audio. The sequence consists of values that are either 1 or -1.



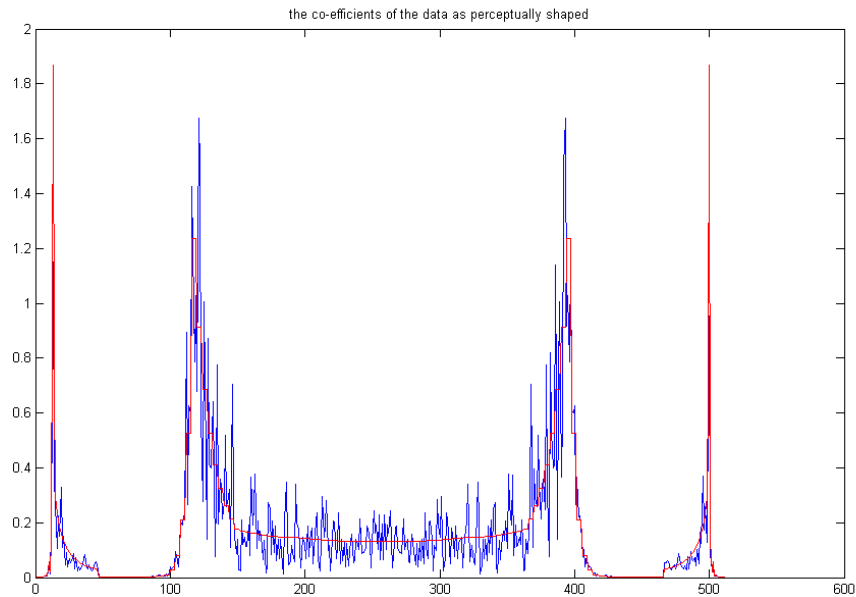
**Figure 5.7: Watermark signal to be inserted in the audio.**

Next, a FFT of the sequence is carried so that the FFT coefficients can be shaped according to the masking thresholds obtained previously. Figure 5.8 shows the FFT of the sequence.



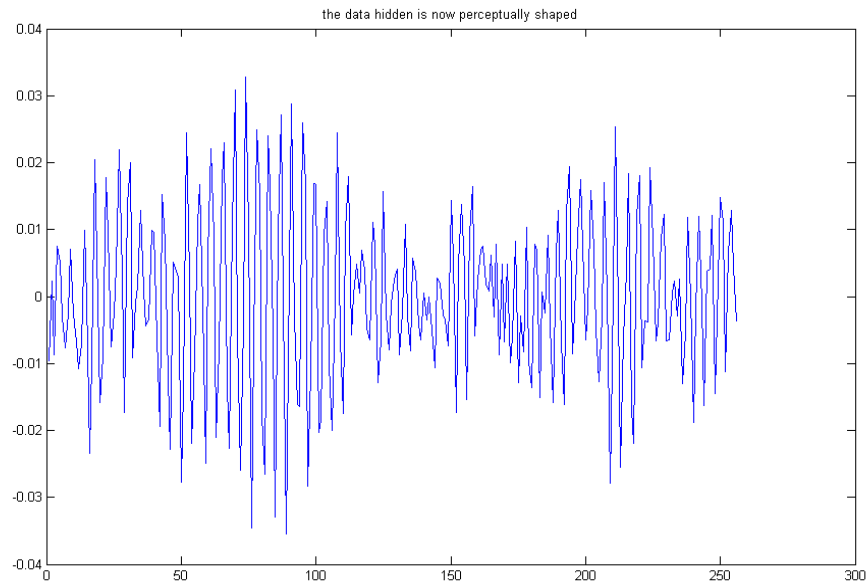
**Figure 5.8: Fast fourier transform of the watermark data.**

The FFT is shaped with the masking threshold. According to experiments the shaping that resulted in most silent watermark was a simple multiplication of the masking threshold and the FFT. Figure 5.9 illustrates this step.



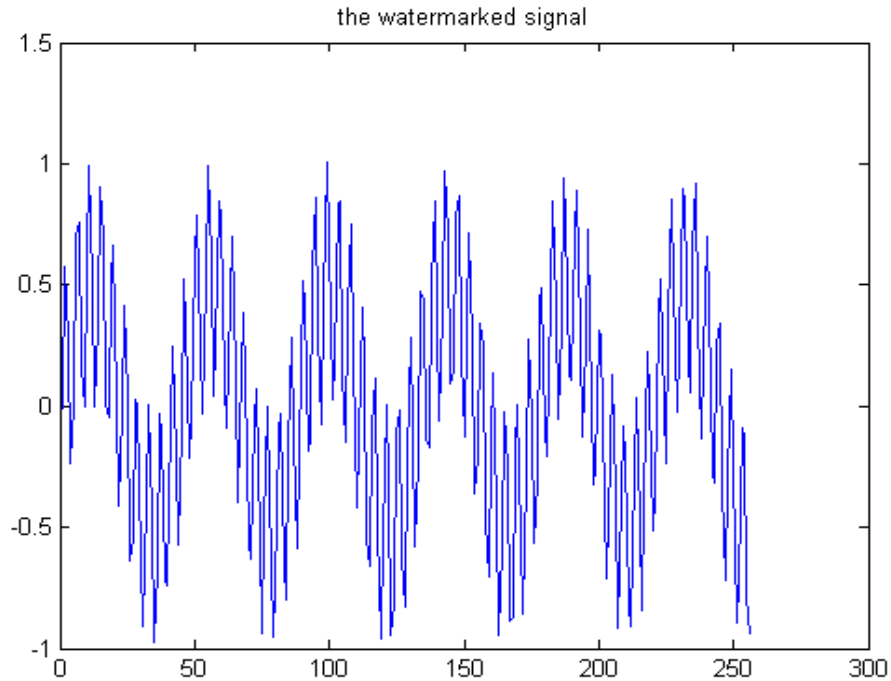
**Figure 5.9: Shaping of the FFT of the watermark.**

The Inverse FFT (IFFT) of the perceptually shaped FFT coefficients, obtained above, is done in order to get the watermark signal to be hidden. Figure 5.10 illustrates the same.



**Figure 5.10: Perceptually shaped watermark obtained by an inverse FFT**

Finally, the watermarked audio is obtained by simple addition of the host audio signal and the shaped watermark. In figure 5.11, we can see the watermarked signal. The presence of watermark can be clearly seen in the figure.







**Figure 5.11: The final watermarked signal**

### **5.3 Performance of Spread Spectrum with psychoacoustic shaping scheme**

The algorithm was implemented and was tested against MP3 compression attacks. The results obtained are shown in the following table.

**Table 5.1: Performance of SS watermark with psychoacoustic shaping against MP3 compression attack.**

Attack Category: MP3 attack		Watermarking scheme: Spread Spectrum with psychoacoustic shaping				
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
MP3 compression, 128kbps	 NC = 0.4989	 NC = 0.4845	 NC = 0.4812	 NC = 0.5121	 NC = 0.4570	 NC = 0.5044
MP3 compression, 64kbps	 NC = 0.5243	 NC = 0.4669	 NC = 0.5011	 NC = 0.4956	 NC = 0.4834	 NC = 0.5121



CHAPTER 6

**DISCRETE WAVELET**

**TRANSFORM**

## 6 *Discrete Wavelet Transform*

---

The wavelet transform is a new tool of signal processing in recent years. Its main character is that it can decompose signals into different frequency components and analyzes signal in the time domain and frequency domain simultaneously. So the wavelet transform is used widely in many fields of signal processing. In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled.

The first DWT was invented by the Hungarian mathematician **Alfréd Haar**. Now, it is known as a DWT with haar filter. In our experiments, we have used the ‘haar’ filter because of its simplicity of application and impeccable performance. For an input represented by a list of  $2n$  numbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale: finally resulting in  $2n - 1$  differences and one final sum.

This simple DWT illustrates the desirable properties of wavelets in general. First, it can be performed in  $O(n)$  operations; second, it captures not only a notion of the frequency content of the input, by examining it at different scales, but also temporal content, i.e. the times at which these frequencies occur. Combined, these two properties make the Fast wavelet transform (FWT), an alternative to the conventional Fast Fourier Transform (FFT).

The most commonly used set of discrete wavelet transforms was formulated by the Belgian mathematician Ingrid Daubechies in 1988. This formulation is based on the use of recurrence relations to generate progressively finer discrete samplings of an implicit mother wavelet function; each resolution is twice that of the previous scale. In her seminal paper, Daubechies derives a family of wavelets, the first of which is the Haar wavelet. Interest in this field has exploded since then, and many variations of Daubechies' original wavelets were developed.

Other forms of discrete wavelet transform include the non- or undecimated wavelet transform (where downsampling is omitted), the Newland transform (where an orthonormal basis of wavelets is formed from appropriately constructed top-hat filters in frequency space). Wavelet packet transforms are also related to the discrete wavelet transform. Complex wavelet transform is another form.

The discrete wavelet transform has a huge number of applications in science, engineering, mathematics and computer science. Most notably, it is used for signal coding, to represent a discrete signal in a more redundant form, often as a preconditioning for data compression.

## 6.1 Mathematical Definition

### 6.1.1 One level of the transform

The DWT of a signal  $x$  is calculated by passing it through a series of filters. First the samples are passed through a low pass filter with impulse response  $g$  resulting in a convolution of the two:

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k]. \quad \dots\dots\dots (6.1)$$

The signal is also decomposed simultaneously using a high-pass filter  $h$ . The outputs giving the detail coefficients (from the high-pass filter) and approximation coefficients (from the low-pass). It is important that the two filters are related to each other and they are known as a quadrature mirror filter.

However, since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist's rule. The filter outputs are then downsampled by 2 (It should be noted that Mallat's and the common notation is the opposite,  $g$ - high pass and  $h$ - low pass):

$$y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k] \quad \dots\dots\dots (6.2)$$

$$y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n - k] \quad \dots\dots\dots (6.3)$$

This decomposition has halved the time resolution since only half of each filter output characterises the signal. However, each output has half the frequency band of the input so the frequency resolution has been doubled. Figure 6.1 illustrates a simple block diagram of a general filter for a discrete wavelet transform.



**Figure 6.1: Block diagram of filter analysis.**

With the downsampling operator  $\downarrow$ ,

$$(y \downarrow k)[n] = y[kn] \quad \dots\dots\dots (6.4)$$

the above summation can be written more concisely.

$$y_{\text{low}} = (x * g) \downarrow 2 \quad \dots\dots\dots (6.5)$$

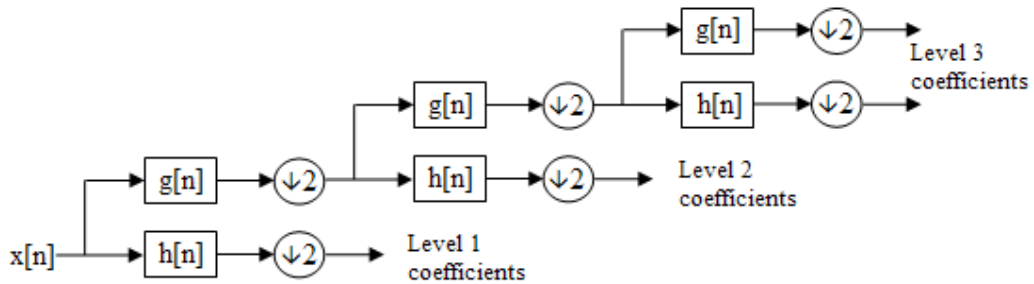
$$y_{\text{high}} = (x * h) \downarrow 2 \quad \dots\dots\dots (6.6)$$

However computing a complete convolution  $x * g$  with subsequent down-sampling would waste computation time.

### 6.1.2 Cascading and Filter banks

This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with

a different time-frequency localisation. The tree is known as a filter bank. Figure 6.2 illustrates a 3-level filter bank.

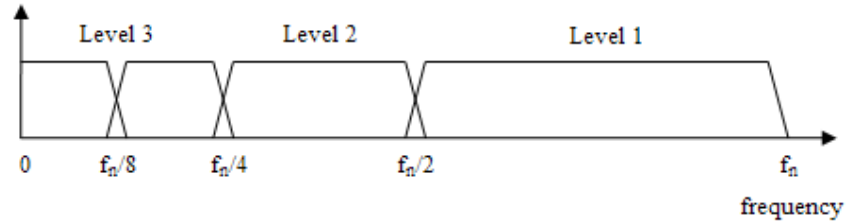


**Figure 6.2: A 3 level filter bank.**

At each level in the above diagram the signal is decomposed into low and high frequencies. Due to the decomposition process the input signal must be a multiple of  $2^n$  where  $n$  is the number of levels. For example a signal with 32 samples, frequency range 0 to  $f_n$  and 3 levels of decomposition, 4 output scales are produced. Figure 6.3 represents this in the form of a diagram.

**Table 6.1: Table showing the result of cascaded DWT.**

Level	Frequencies	Samples
3	0 to $f_n / 8$	4
	$f_n / 8$ to $f_n / 4$	4
2	$f_n / 4$ to $f_n / 2$	8
1	$f_n / 2$ to $f_n$	16



**Figure 6.3: Frequency domain representation of the DWT.**

## 6.2 Watermarking Algorithms Based on DWT

The principle objective of the wavelet transform is to hierarchically decompose an input signal into a series of low frequency approximation sub-band and their detail sub-bands. For the dyadic wavelet decomposition, at each level, the low frequency approximation sub-band and detail sub-band (or sub-bands for multidimensional case) contain the information needed to reconstruct the low frequency approximation signal at the next higher resolution level. A large number of algorithms have been developed that carry out some levels of cascaded DWT on the host audio and then insert the watermark in one of the partitions obtained. Few of the algorithms that are commonly used are described below.

### 6.2.1 LSB insertion

Data hiding in the LSBs of the wavelet coefficients was practicable due to near perfect reconstruction properties of the filter-bank. The algorithm usually followed is as follows. Signal decomposition into low-pass and high pass part of the spectrum is performed in five successive steps. After sub-band decomposition of 512 samples of host audio, ‘Haar’ filter and decomposition depth of five steps, 512 wavelet coefficients are produced. All 512 wavelet coefficients are then scaled using the maximum value inside the given sub-band and converted to binary arrays in the two’s complement. A predetermined number of the LSBs are thereupon replaced with bits of information that should be hidden inside the host audio. Coefficients are then converted and scaled back to the original order of magnitude and inverse transformation is performed.

### 6.2.2 Mean Quantization in DWT

In this scheme of watermarking, quantization of the coefficients obtained after some levels of DWT are used for embedding. Low-frequency coefficients are selected to embed watermark using mean-quantization method i.e. quantization of the value of the mean in order to denote a bit 0 or a bit 1, in order to guarantee the robustness of the watermark. Assume that transform domain coefficients make up of an aggregate of  $\{x_0, x_1, \dots, x_{k-1}\}$  and its mean is calculated by  $\bar{x} = \frac{1}{K} \sum_{i=0}^{K-1} x_i$ . Assume that embedding one watermark bit of  $w_i \in \{0, 1\}$  using quantization method in  $\bar{x}$  causes an error and the error is  $\Delta$ , so the mean is  $x^* = \bar{x} + \Delta$  after embedding one watermark bit. The value  $\Delta$  is such that  $x^*$  denotes whether the value to be hidden is a 1 or a 0.

CHAPTER 7

**SPREAD SPECTRUM**

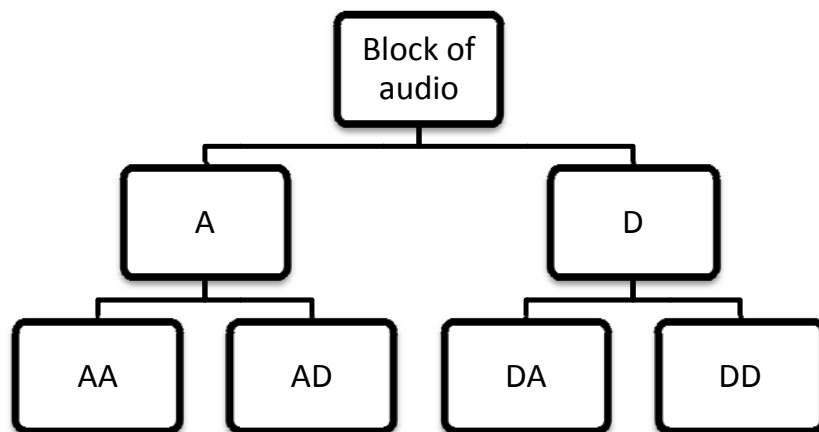
**AFTER DWT**



## 7 Spread Spectrum after DWT

---

In our endeavor to develop a robust and blind watermarking method, the DWT tool seemed very interesting and attractive. The advantage of discrete wavelet transform over other transforms like FFT etc is that if K levels of discrete wavelet transform of a host signal is carried out, then each value of the host signal gets distributed into values. To make this clearer, let us carry out two levels of DWT on a block of original host audio as shown in figure 7.1.



**Figure 7.1: Figure showing two levels of DWT**

Now according to theory, each value of the block of audio is divided into the four blocks AA, AD, DA and DD. It means that during inverse discrete wavelet transform, to calculate each element of the audio, all the four blocks are essentially required. In other words, each value of the original signal is dependent on values from all the four blocks.

Based on this concept, we anticipated that if an embedding similar to spread spectrum is done in one of the blocks after few levels of DWT, then the watermark will spread into a large block of the audio in a complex way (dependent of which transform filter we use). This spreading and complexity of the transform are the factors which we thought will improve the robustness of spread spectrum watermarking.

## 7.1 Performance of spread spectrum after DWT scheme













In our experiments, we first do a two-level discrete wavelet transform to get AA, AD, DA and DD components as described in figure 7.1. Then one of any of the four divisions is taken and embedding similar to spread spectrum is done in it. Inverse discrete wavelet transform is computed to obtain the watermarked audio. The results for embedding in the DD and AA divisions are shown in the following pages.

### 7.1.1 Embedding in section DD

























Table 7.1: SNR and PSNR values obtained after embedding

File name	SNR	PSNR
Piano.wav	21.7338	38.0621
Orchestra.wav	15.1412	39.4656
Beats.wav	20.3044	40.2859
Song.wav	13.2534	38.0621
Pop.wav	12.1685	40.8265
Jazz.wav	18.1213	38.0621







Table 7.2: Performance of spread spectrum after DWT in section DD towards MP3 attacks.

Attack Category: MP3 attack		Watermarking scheme: spread spectrum after DWT in section DD				
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
MP3 compression, 128kbps	 NC = 0.8720	 NC = 0.7450	 NC = 0.9095	 NC = 0.7969	 NC = 0.8940	 NC = 0.8057
MP3 compression, 64kbps	 NC = 0.8720	 NC = 0.8157	 NC = 0.7914	 NC = 0.7031	 NC = 0.7329	 NC = 0.8455

**Table 7.3: Performance of spread spectrum after DWT in section DD towards re-sampling attacks.**

Attack Category: Re-sampling attack			Watermarking scheme: spread spectrum after DWT in section DD			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Re-sampling factor = 2	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000
Re-sampling factor = 3	 NC = 0.9956	 NC = 0.9956	 NC = 1.0000	 NC = 1.0000	 NC = 0.9801	 NC = 0.9945
Re-sampling factor = 4	 NC = 0.8885	 NC = 0.8830	 NC = 0.9934	 NC = 1.0000	 NC = 0.9834	 NC = 0.9360
Re-sampling factor = 5	 NC = 0.7461	 NC = 0.7737	 NC = 0.9547	 NC = 0.9812	 NC = 0.9393	 NC = 0.8576

**Table 7.4: Performance of spread spectrum after DWT in section DD towards white noise attacks**













Attack Category: White Noise addition			Watermarking scheme: spread spectrum after DWT in section DD			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Gaussian Noise	 NC = 0.9890	 NC = 0.9845	 NC = 0.9912	 NC = 0.9890	 NC = 0.9879	 NC = 0.9879

### 7.1.2 Embedding in section AA

























**Table 7.5: SNR and PSNR values obtained after embedding**

File name	SNR	PSNR
Piano.wav	21.7338	32.0415
Orchestra.wav	15.1412	39.4656
Beats.wav	20.3044	40.2859
Song.wav	13.2534	38.0621
Pop.wav	12.1685	40.8265
Jazz.wav	18.1213	38.0621







**Table 7.6: Performance of spread spectrum after DWT in section AA towards MP3 attacks.**

Attack Category: MP3 attack			Watermarking scheme: spread spectrum after DWT in section AA			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
MP3 compression, 128kbps	 NC = 0.4967	 NC = 0.5717	 NC = 0.5607	 NC = 0.4227	 NC = 0.4746	 NC = 0.5088
MP3 compression, 64kbps	 NC = 0.5232	 NC = 0.5210	 NC = 0.4536	 NC = 0.5519	 NC = 0.4172	 NC = 0.4393

**Table 7.7: Performance of spread spectrum after DWT in section AA towards re-sampling attacks.**

Attack Category: Re-sampling attack			Watermarking scheme: spread spectrum after DWT in section AA			
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Re-sampling factor = 2	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 0.9912
Re-sampling factor = 3	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 1.0000	 NC = 0.9967	 NC = 0.9812
Re-sampling factor = 4	 NC = 0.9978	 NC = 0.9989	 NC = 0.9945	 NC = 1.0000	 NC = 0.9967	 NC = 0.9360
Re-sampling factor = 5	 NC = 0.9945	 NC = 0.9989	 NC = 0.9868	 NC = 0.9812	 NC = 0.9934	 NC = 0.9669

**Table 7.8: Performance of spread spectrum after DWT in section AA towards white noise attacks**

Attack Category: White Noise addition		Watermarking scheme: spread spectrum after DWT in section AA				
Attack Detail	Piano.wav	Orchestra.wav	Beats.wav	Song.wav	Pop.wav	Jazz.wav
Gaussian Noise	 NC = 0.9735	 NC = 0.9823	 NC = 0.9845	 NC = 0.9890	 NC = 0.9857	 NC = 0.9558

## *8 Conclusions and Future Work*

---

### **8.1 Summary of Our Work**

From the experiments performed, we conclude that shaping of the psychoacoustic shaping of spread spectrum watermark even though produces a perceptually transparent watermarking scheme, still it fails against simple watermarking attacks such as MP3 compression. However, when spread spectrum is done after DWT, robustness increases due to the mathematical computations involved in the transform. It can be easily observed from the tables that watermark insertion in section DD (refer figure 7.1) is more robust to MP3 attacks. On the other hand the scheme involving insertion in AA segment is more robust to re-sampling attack. We suggest an insertion of watermark in both the segments in ratio of strengths in order to in order to obtain a watermark robust to both MP3 compression attacks and re-sampling attacks.

### **8.2 Future Scope of Work**

Although we have achieved a watermarking scheme robust to intelligent attacks like dual watermarking, as well as simple signal manipulation attacks like MP3 compression, re-sampling and white noise addition. Still there remains a scope of improvement. According to results, different files give different performance to the same watermarking scheme. A closer look into the audio files that have been used may reveal us certain properties that an audio file must possess in order to give best results to our watermarking scheme. Similarly, we may also formulate rules to identify certain regions in each audio file that will give best robustness when our watermark is embedded and hence we may only in those regions. As technology advances, more attacks may come into existence and we may need to incorporate features to combat them.

## *Bibliography*

---

- [1] Nedeljko Cvejic, "ALGORITHMS FOR AUDIO WATERMARKING AND STEGANOGRAPHY," Oulu 2004.
- [2]. Darko Kirovski and Henrique S. Malvar, "Spread-Spectrum Watermarking of Audio Signals," in IEEE transactions on Signal Processing , VOL.51, NO.4, APRIL2003.
- [3] NedeGko Cvejic, Tapio Seppanen, "A wavelet domain LSB insertion algorithm for high capacity audio steganography," IEEE 2002.
- [4] Mitchell D Sanson, Bin Zhu, Ahmed H Tewk and Laurence Boney, "Robust audio watermarking using perceptual masking," 1996.
- [5] P. Bassia and I. Pitas, "Robust audio watermarking in the time domain," Proc. EUSIPCO 98, ol. 1, pp. 25–28, Rodos, Grece, Sept. 1998.
- [6] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in Information Hiding, Springer Lecture Notes in Computer Science, v1174, pp. 295–315, 1996.
- [7] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "A secure robust watermark for multimedia," information Hiding Worshop, Univ. of Cambridge, pp.185–206, 1996.
- [8] C. Neubauer and J. Herre, "Digital watermarking and its influence on audio quality," Proc. 105th Convention, Audio Engineering Society, San Francisco, CA, Sept. 1998.
- [9] M.D. Swanson, B. Zhu, A.H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," Signal Processing, vol.66, pp. 337–355, 1998.
- [10] B. Chen and G. W. Wornell, "Digital watermarking and Information embedding using dither modulation," Proc. IEEE Workshop on Multimedia Signal Processing, Redondo Beach, Cpp. 273–278, Dec. 1998.

- [11] Tang Xianghong, Niu Yamei, Li Qiliang, "A digital audio watermark embedding algorithm with wt and cct," IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings, 2005.
- [12] Ming Li, Yun Lei, Jian Liu and Yonghong Yan, "A novel audio watermarking in wavelet domain," Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006.
- [13] Liu Hai-yan, Zheng Xue-feng and Wang Ying, "dwt –based audio watermarking resistant to desynchronization," Proc. IEEE 2007.
- [14] Bilge Gonsel and Serap Kirbiz, "Perceptual Audio Watermarking by Learning in Wavelet Domain," 18<sup>th</sup> International Conference on Pattern Recognition 2006.
- [15] Darko Kirovski and Henrique S. Malvar, "Spread-Spectrum Watermarking of Audio Signals," IEEE 2003.
- [16] Recording Industry Association of America [Online]. Available: <http://www.riaa.org>.
- [17] D. Kirovski, H. S. Malvar, and Y. Yacobi. (2001) A dual watermarking and fingerprinting system. Microsoft Research. [Online]. Available: <http://research.microsoft.com>.
- [18] S. Katzenbeisser and F. A. P. Petitcolas, Hiding Techniques for Steganography and Digital Watermarking, S. Katzenbeisser and F. A. P. Petitcolas, Eds. Boston, MA: Artech House, 2000.
- [19] P. Bassia and I. Pitas, "Robust audio watermarking in the time domain," in Proc. EUSIPCO, vol. 1, Rodos, Greece, Sept. 1998, pp. 25–28.
- [20] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "A secure, robust watermark for multimedia," in Proc. Inform. Hiding Workshop, Cambridge, U.K., June 1996, pp. 147–158.



- [21] C. Neubauer and J. Herre, "Digital watermarking and its influence on audio quality," in Proc. 105th AES Conv., San Francisco, CA, Sept. 1998.
- [22] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Process.*, vol. 66, no. 3, pp. 337–355, 1998.
- [23] D. Gruhl, A. Lu, and W. Bender, "Echo hiding," in Proc. Inform. Hiding Workshop, Cambridge, U.K., June 1996, pp. 293–315.
- [24] W. Szepanski, "A signal theoretic method for creating forgery-proof documents for automatic verification," in Proc. Carnahan Conf. Crime Countermeasures, Lexington, KY, May 1979, pp. 101–109.
- [25] B. Chen and G.W. Wornell, "Digital watermarking and information embedding using dither modulation," in Proc. Workshop Multimedia Signal Process., Redondo Beach, CA, Dec. 1998, pp. 273–278.
- [26] J. K. Su and B. Girod, "Power-spectrum condition for energy-efficient watermarking," in Proc. Int. Conf. Image Process., Kobe, Japan, Oct. 1999, pp. 301–305.
- [27] J. P. Linnartz and M. van Dijk, "Analysis of the sensitivity attack against electronic watermarks in images," in Proc. Inform. Hiding Workshop, Portland, OR, Apr. 1998, pp. 258–272.