**Chapter 1**

**INTRODUCTION**

1.1 INTRODUCTION

Development of internet is the root cause behind distributing digital media on the network recently. More information is transmitted in a digital format now than ever, and the growth in this trend will not plateau in the foreseeable future. Digital information is susceptible to having copies made at the same quality as the original. As a new solution for content authentication, digital watermarking, is drawing considerable attention and becomes an active research field. A watermarking algorithm consists of the watermark structure [1], an embedding algorithm, and an extraction, or detection, algorithm. As a new solution for content authentication, digital watermarking, is drawing considerable attention and becomes an active research field. A watermarking algorithm consists of the watermark structure, an embedding algorithm, and an extraction, or detection, algorithm.

Digital Watermarking [40] is one of the most powerful techniques that has come into focus for safeguarding multi- media contents like audio, video and images against copyright violations. We have focused on audio signals where copyright infringement, piracy, etc. become a major issue. Digital audio watermarking is an important information hiding technique that embeds some additional information called watermarks into the audio file which can be used to trace and verify the authenticity and legal use of the concerned audio file. It also helps in ascertaining genuine ownerships in the court of law.

Watermarks can be broadly classified into two types, perceptible and imperceptible. Perceptible watermarks are visual patterns that are imposed on images and videos. Imperceptible watermarks are invisible patterns that are unrecognized by Human Auditory System (HAS) and Human Visual System (HVS), and can only be detected algorithmically. Watermarks can also be classified as robust or fragile depending on whether they are able to withstand various signal processing attacks. Here a robust audio watermarking scheme has been proposed which is able to resist many signal processing attacks.

Compared to embedding watermarks into images, audio watermarking is a more challenging task due to the fact that the human auditory system (HAS) is more sensitive to distortions than the human visual system (HVS) [1], and that inaudibility is much more difficult to achieve than invisibility for images [2]. Also, compared to the visual signals, audio signals are represented by much less number of samples per time interval, which limits the watermark capacity for the audio signals. Several techniques in audio watermarking have been proposed to address these challenges, including the echo coding

A few audio watermark techniques have been reported [2]. Especially it is hard to find to find robust audio watermark algorithm which can resist the synchronization attack effectively [3]-[7].

Synchronization attack is referred as changing the embedding position of the watermark so that detector cannot detect the right watermark. Kim et al. [8] proposed a robust audio watermarking strategy using common binary sequence as synchronization, but the relatively poor periodic and aperiodic relativity of common binary sequence weakens the ability of resist synchronization attack. Barkers code has better self-relativity, so Wang [9] used 16 bit barkers code as synchronization mark and is embedded in modifying several mean value of several samples. The watermark is embedded into DWT and DCT domain. But it has defect that the scheme fails in case of mp3 compression and noise addition.

Taking the above mentioned problem into consideration we introduced a DWT and DCT based watermarking algorithm which use the scheme propose by Wang [10] combined with energy efficient watermarking technique using energy efficient watermark as proposed by Jonathan [11], which is to generate a watermark that satisfy the Power Spectrum Condition (PSC). The watermark generated through this technique is highly robust and can resist all attack. As the scheme proposed by Jonathan is a non-blind technique the original audio file is required for extraction and it also fail for the cropping attack. So the scheme we introduced is combination of a blind and non-blind technique and they compensate each other drawbacks. The combined scheme is highly robust.

1.2 Organization of Dissertation

This report will begin with a quick background on cryptography and steganography, which form the basis for a large number of digital watermarking concepts. Then it will move on to a discussion of what requirements a watermarking system must meet, as well as methods for evaluating the strengths of various algorithms. The remainder of the report will focus on various watermarking techniques. This report will focus almost exclusively on the watermarking of digital audio file; however most of these same ideas could easily be applied to the watermarking of digital video and images.

Chapter 1 gives the introduction about the digital audio watermarking; it explains the basic watermarking related to this project. This chapter explains about the crypto-graphical techniques and information hiding. The last part gives the structure of report.

Chapter 2 is concerned about the review of the related literature required for the project. This chapter gives the history of watermarking, information hiding, and stenography. It also includes distortions and attacks on images. The last part of the Chapter 2 gives an overview of applications of the digital watermarking techniques and design consideration and requirements.

Chapter 3 gives the basic idea about audio watermarking techniques, purpose and attribute of watermark image. It also includes spatial, frequency and wavelet domain and some other audio watermarking techniques used and there advantages and disadvantages.

Chapter 4 describes the Fundamental theory and the proposed algorithm of the Audio watermarking. It includes various type of Audio file format, need of Synchronization code and barkers code used for synchronization. Quantization method and Cox algorithm used for embedding Dual Audio Watermark and algorithm of the proposed watermarking scheme.

Chapter 5 shows the running layout of this project with describing each and every Mat Lab generated figures and corresponding watermark detector response. It describes the result of various attack including Mp3 compression, requantization, resampling, cropping, noise addition and the detector response for all attacks.

Chapter 6 discusses about the conclusion and future directions in this project.

Chapter 7 shows the references which is used for making this report.

**Chapter 2**

**HISTORY OF DIGITAL WATERMARKING**

2.1 HISTORY OF WATERMARKING

The art of papermaking was invented in China over one thousand years earlier, paper watermarks did not appear until about 1282, in Italy [12]. The marks were made by adding thin wire patterns to the paper molds. The paper would be slightly thinner where the wire was and hence more transparent. The meaning and purpose of the earliest watermarks are uncertain. They may have been used for practical functions such as identifying the molds on which sheets of papers were made, or as trademarks to identify the papermaker. On the other hand, they may have represented mystical signs, or might simply have served as decoration.

It is difficult to determine when digital watermarking was first discussed. In 1979, Szepanski described a machine-detectable pattern that could be placed on documents for anti-counterfeiting purposes. Nine years later, Holt described a method for embedding an identification code in an audio signal. However, it was Komatsu and Tominaga in 1988, which appear to have first used the term digital watermark. Still, it was probably not until the early 1990s that the term digital watermarking really came into vogue. About 1995, interest in digital watermarking began to mushroom.

Watermarking is descendent of steganography which has been in existence for at least a few hundred years. Watermarking is a special technique of steganography where one message is embedded in another and the two messages are related to each other in some way. The most common examples of watermarking are the presence of specific patterns in currency notes which are visible only when the note is held to light and logos in the background of printed text documents. The watermarking techniques prevent forgery and unauthorized replication of physical objects.

Digital watermarking is similar to watermarking physical objects except that the watermarking technique is used for digital content instead of physical objects. In digital watermarking a low-energy signal is imperceptibly embedded in another signal. The low-energy signal is called watermark and it depicts some metadata, like security or rights information about the main signal. The main signal in which the watermark is embedded is referred to as cover signal since it covers the watermark. The cover signal is generally a still image, audio clip, video sequence or a text document in digital format.

Unlike encryption, which does not provide a way to examine the original data in its protected form, the watermark remains in the content in its original form and does not prevent a user from listening to, viewing, examining, or manipulating the content. Also, unlike the idea of steganography, where the method of hiding the message may be secret and the message itself is secret, in watermarking, typically the watermark embedding process is known and the message (except for the use of an optional secret key) does not have to be secret.

Watermarking is the direct embedding of additional information into the original content or host signal. Ideally, there should be no perceptible difference between the watermarked and original signal and the watermark should be difficult to remove or alter without damaging the host signal. In some instances, the amount of information that can be hidden and detected reliably is important. It is easy to see that the requirements of imperceptibility, robustness, and capacity conflict with each other. For instance, a straightforward way to provide an imperceptible watermark is to embed the watermark signal into the perceptually insignificant portion of the host data. However, this makes the watermark vulnerable to attack because it is fairly easy to remove or alter the watermark without affecting the host signal.

To provide a robust watermark, a good strategy is to embed the watermark signal into the significant portion of the host signal. This portion of the host data is highly sensitive to alterations, however, and may produce very audible or visible distortions in the host data. Applications for digital watermarking include copyright protection, fingerprinting, authentication, copy control, tamper detection, and data hiding applications such as broadcast monitoring. Watermarking algorithms have been developed for audio, still images, video, graphics, and text.

Visible watermarks which do not interfere with the intelligibility of the host signal have also been developed; while transparent watermarking techniques can be fragile, robust, or semi fragile. Fragile watermarks do not survive lossy transformations to the original host signal and their purpose is tamper detection of the original signal.

There are many effective ways to insert a fragile watermark into digital content while preserving the imperceptibility requirement. Placing the watermark information into the perceptually insignificant portions of the data guarantees imperceptibility and provides fragile marking capabilities. For instance, early watermark techniques for still image data propose inserting watermark information into the least significant bits of the pixel values. These results in an imperceptible mark which can detect lossy transformations performed on the watermarked content. For security applications and copyright protection, robust watermarking techniques have been developed. Here the technical challenge is to provide transparency and robustness which are conflicting requirements.

Ideally, an effective, robust watermarking scheme provides a mark that can only be removed when the original content is destroyed as well. The degree of robustness and distortion necessary to alter the value of the original content can vary for different applications. Typically, many of the applications for copyright protection involve relatively high quality original content and the imperceptibility criterion is critical for such applications. In order for a watermarking technique to be robust, the watermark should be embedded in the perceptually significant portion of the data.

Some typical distortions or attacks that digital watermarking schemes are expected to survive include re-sampling, rescaling, compression, linear and nonlinear filtering, additive noise, A/D and D/A conversion, and trans-coding. Applications for robust watermarking include copyright protection where each copy gets a unique watermark (commonly referred to as a fingerprint) to identify the end-user so that tracing is possible for cases of illegal use; authentication, where the watermark can represent a signature and copy control for digital recording devices. Within the class of robust watermarking techniques there are several different constraints on encoder and decoder design which depends on the particular application.

Semi-fragile watermarking techniques differentiate between lossy transformations that are “information preserving” and lossy transformations which are “information altering.” Lossy transformations include any signal processing step that alters the original signal values and is not invertible. For example, in authentication applications it may be desirable to have a watermark that can distinguish between lossy transformations such as compression which does not alter the integrity of the content and an alteration which does alter the integrity, such as manipulating or replacing objects within the scene.

2.2 CRYPTOGRAPHY

Literally, Cryptography is the art of writing in ‘ciphers’; or it is a method of secret communication. In cryptography, the contents of secret message are concealed and only the sender and the receiver of the secret message know the process of extracting the concealed information. Apparently, others can’t easily detect what message is being conveyed. Cryptography is an effective solution to the distribution problem, but in most instances has to be tied to specialized and costly hardware to create tamper-proof devices that avoid direct access to data in digital format. Moreover, most cryptographic protocols are concerned with secured communications instead of ulterior copyright infringements. For instance, access control in set-top-boxes used for digital television demodulation and decoding succeed in avoiding unauthorized access to programs that are being broadcast in scrambled form but fail in precluding further storage and illegal dissemination actions.

2.3 STEGANOGRAPHY

Steganography is derived from the Greek for covered writing and essentially means “to hide in plain sight”. Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have been in use for hundreds of years, but with the increasing use of files in an electronic format new techniques for information hiding have become possible.

Stenography improves on this by hiding the fact that a communication even occurred. The message m is imbedded into a harmless message c which is defined as the cover object. The message m is then embedded into c, generally with use of a key k that is defined as the stego-key. The resulting message is then embedded into the cover-object c, which results in stego-objects. Ideally the stego-object is indistinguishable from the original message c, appearing as if no other information has been encoded [13]. This can all be seen below in figure 2.1.



Figure 2.1 Illustration of a Stegnographic System

2.4 Digital Watermarking

Digital watermarking is the process of embedding information into a digital signal. The signal may be audio, pictures or video, for example. If the signal is copied, then the information is also carried in the copy.

In visible watermarking, the information is visible in the picture or video. Typically, the information is text or a logo which identifies the owner of the media. When a television broadcaster adds its logo to the corner of transmitted video, this is also a visible watermark.

In invisible watermarking, information is added as digital data to audio, picture or video, but it cannot be perceived as such. An important application of invisible watermarking is to copyright protection systems, which are intended to prevent or deter unauthorized copying of digital media.

Unlike encryption, which is useful for transmission but does not provide a way to examine the original data in its protected form, the watermark remains in the content in its original form and does not prevent a user from listening to, viewing, examining, or manipulating the content. Also, unlike the idea of steganography, where the method of hiding the message may be secret and the message itself is secret, in watermarking, typically the watermark embedding process is known and the message (except for the use of a secret key) does not have to be secret. In steganography, usually the message itself is of value and must be protected through clever hiding techniques and the “vessel” for hiding the message is not of value. In watermarking, the effective coupling of message to the “vessel,” which is the digital content, is of value and the protection of the content is crucial. Watermarking is the direct embedding of additional information into the original content or host signal. Ideally, there should be no perceptible difference between the watermarked and original signal [14], [15] and the watermark should be difficult to remove or alter without damaging the host signal. In some instances, the amount of information that can be hidden and detected reliably is important. It is easy to see that the requirements of imperceptibility, robustness, and capacity conflict with each other. For instance, a straightforward way to provide an imperceptible watermark is to embed the watermark signal into the perceptually insignificant portion of the host data. However, this makes the watermark vulnerable to attack because it is fairly easy to remove or alter the watermark without affecting the host signal. To provide a robust watermark, a good strategy is to embed the watermark signal into the significant portion of the host signal. This portion of the host data is highly sensitive to alterations, however, and may produce very audible or visible distortions in the host data. Applications for digital watermarking include copyright protection, fingerprinting, authentication, copy control, tamper detection, and data hiding applications such as broadcast monitoring.



Figure 2.2 Block diagram of a watermarking system

Watermarking is defined as the practice of imperceptibly altering a Work to embed a message about that Work and Steganography is defined as the practice of undetectably altering a Work to embed a secret message. Figure 2.2 shows the basic block diagram of a watermarking system.

2.5 DISTORSIONS AND ATTACKS

In practice, a watermarked object may be altered either on purpose or accidentally, so the watermarking system should still be able to detect and extract the watermark. Obviously, the distortions are limited to those that do not produce excessive degradations, since otherwise the transformed object would be unusable. These distortions also introduce degradation on the performance of the system. For intentional attacks, the goal of the attacker is to maximize the reduction in these probabilities while minimizing the impact that his/her transformation produces on the object; this has to be done without knowing the value of the secret key used in the watermarking insertion process, which is where all the security of the algorithm lies. Next, we introduce some of the best known attacks. Some of them may be intentional or unintentional, depending on the application:

2.5.1 Additive Noise

This may stem in certain applications from the use of D/A and A/D converters or from transmission errors. However, an attacker may introduce perceptually shaped noise (thus, imperceptible) with the maximum unnoticeable power. This will typically force to increase the threshold at which the correlation detector works.

2.5.2 Filtering

Low-pass filtering, for instance, does not introduce considerable degradation in watermarked images or audio, but can dramatically affect the performance, since spread-spectrum-like watermarks have non negligible high-frequency spectral contents.

2.5.3 Cropping

This is a very common attack since in many cases the attacker is interested in a small portion of the watermarked object, such as parts of a certain picture or frames of a video sequence. With this in mind, in order to survive, the watermark needs to be spread over the dimensions where this attack takes place.

2.5.4 Compression

This is generally an unintentional attack which appears very often in multimedia applications. Practically all the audio, video and images that are currently being distributed via Internet have been compressed. If the watermark is required to resist different levels of compression, it is usually advisable to perform the watermark insertion task in the same domain where the compression takes place. For instance, DCT domain image watermarking is more robust to JPEG compression than spatial-domain watermarking.

2.5.5 Rotation and Scaling

This has been the true battle horse of digital watermarking, especially because of its success with still images. Correlation based detection and extraction fail when rotation or scaling is performed on the watermarked image because the embedded watermark and the locally generated version do not share the same spatial pattern anymore. Obviously, it would be possible to do exhaustive search on different rotation angles and scaling factors until a correlation peak is found, but this is prohibitively complex. Note that estimating the two parameters become simple when the original image is present, but we have argument against this possibility in previous sections. In [10] the authors have shown that although the problem resembles synchronization for digital communications, the techniques applied there fail loudly. Some authors have recently proposed the use of rotation and scaling-invariant transforms (such as the Fourier-Mellin [11]) but this dramatically reduces the capacity for message hiding. In any case, publicly available programs like Strirmark break the uniform axes transformation by creating an imperceptible non-linear resampling of the image [9] that renders invariant transforms unusable. In audio watermarking it is also quite simple to perform a non-linear transformation of the time axis that considerably difficult watermark detection.

2.5.6 Statistical Averaging

An attacker may try to estimate the watermark and then ‘unwatermark’ the object by subtracting the estimate. This is dangerous if the watermark does not depend substantially on the data. Note that with different watermarked objects it would be possible to improve the estimate by simple averaging. This is a good reason for using perceptual masks to create the watermark.

2.5.7 Multiple Watermarking

An attacker may watermark an already watermarked object and later make claims of ownership. The easiest solution is to timestamp the hidden information by a certification authority.

2.5.8 Attacks at Other Levels

There are a number of attacks that are directed to the way the watermark is manipulated. For instance, it is possible to circumvent copy control mechanisms discussed below by super scrambling data so that the watermark is lost or to deceive web crawlers searching for certain watermarks by creating a presentation layer that alters the way data are ordered. The latter is sometimes called ‘mosaic attack’ [11].

2.6 APPLICATIONS OF DIGITAL WATERMARKING

In this section we discuss some of the scenarios where watermarking is being already used as well as other potential applications. The list given here is by no means complete and intends to give a perspective of the broad range of business possibilities that digital watermarking opens.

2.6.1 Video Watermarking

In this case, most of the considerations made in previous sections hold. However, now the temporal axis can be exploited to increase the redundancy of the watermark. As in the still images case, watermarks can be created either in the spatial or in the DCT domains. In the latter, the results can be directly extrapolated to MPEG-2 sequences, although different actions must be taken for I, P and B frames. Note that perhaps the set of attacks that can be performed intentionally is not smaller but definitely more expensive than for still images.

2.6.2 Audio Watermarking

Again, previous considerations are valid. In this case, time and frequency masking properties of the human ear are used to conceal the watermark and make it inaudible. The greatest difficulty lies in synchronizing the watermark and the watermarked audio file, but techniques that overcome this problem have been proposed.

2.6.3 Hardware/Software Watermarking

This is a good paradigm that allows us to understand how almost every kind of data can be copyright protected. If one is able to find two different ways of expressing the same information, then one bit of information can be concealed, something that can be easily generalized to any number of bits. This is why it is generally said that a perfect compression scheme does not leave room for watermarking. In the hardware context, Boolean equivalences can be exploited to yield instances that use different types of gates and that can be addressed by the hidden information bits. Software can be also protected not only by finding equivalences between instructions, variable names, or memory addresses, but also by altering the order of non-critical instructions. All this can be accomplished at compiler level.

2.6.4 Text Watermarking

This problem, which in fact was one of the first that was studied within the information hiding area, can be solved at two levels. At the printout level, information can be encoded in the way the text lines or words are separated (this facilitates the survival of the watermark even to photocopying). At the semantic level (necessary when raw text files are provided), equivalences between words or expressions can be used, although special care has to be taken not to destruct the possible intention of the author.

2.6.5 Executable Watermarks

Once the hidden channel has been created it is possible to include even executable contents, provided that the corresponding applet is running on the end user side.

2.6.6 Labeling

The hidden message could also contain labels that allow for example to annotate images or audio. Of course, the annotation may also been included in a separate file, but with watermarking it results more difficult to destroy or loose this label, since it becomes closely tied to the object that annotates. This is especially useful in medical applications since it prevents dangerous errors.

2.6.7 Fingerprinting

This is similar to the previous application and allows acquisition devices (such as video cameras, audio recorders, etc) to insert information about the specific device (e.g., an ID number) and date of creation. This can also be done with conventional digital signature techniques but with watermarking it becomes considerably more difficult to excise or alter the signature. Some digital cameras already include this feature.

2.6.8 Authentication

This is a variant of the previous application, in an area where cryptographic techniques have already made their way. However, are two significant benefits that arise from using watermarking: first, as in the previous case, the signature becomes embedded in the message, second, it is possible to create ‘soft authentication’ algorithms that offer a multi-valued ‘perceptual closeness’ measure that accounts for different unintentional transformations that the data may have suffered (an example is image compression with different levels), instead of the classical yes/no answer given by cryptography-based authentication. Unfortunately, the major drawback of watermarking-based authentication is the lack of public key algorithms that force either to put secret keys in risk or to resort to trusted parties.

2.6.9 Copy and Playback Control

The message carried by the watermark may also contain information regarding copy and display permissions. Then, a secure module can be added in copy or playback equipment to automatically extract this permission information and block further processing if required. In order to be effective, this protection approach requires agreements between content providers and consumer electronics manufacturers to introduce compliant watermark detectors in their video players and recorders. This approach is being taken in Digital Video Disc (DVD).

2.6.10 Signaling

The imperceptibility constraint is helpful when transmitting signaling information in the hidden channel. The advantage of using this channel is that no bandwidth increase is required. An interesting application in broadcasting consists in watermarking commercials with signaling information that permits an automatic counting device to assess the number of times that the commercial has been broadcast during a certain period. An alternative to this would require complex recognition software.

2.7 CHARACTERISTIC FEATURES OF WATERMARKING

As mentioned earlier, digital watermarking techniques are useful for embedding metadata in multimedia content. There are alternate mechanisms like using the header of a digital file to store meta-information. However, for inserting visible marks in images & video and for adding information about audio at the beginning or end of the audio clip etc. the digital watermarking technique is appealing, since it provides following main features and does not require out-of-band data as in other mechanisms.

2.7.1 Imperceptibility

The embedded watermarks are imperceptible both perceptually as well as statistically and do not alter the aesthetics of the multimedia content that is watermarked. The watermarks do not create visible artifacts in still images, alter the bit-rate of video or introduce audible frequencies in audio signals. The watermark should be perceptually invisible, or its presence should not interfere with the work being protected.

2.7.2 Robustness

Depending on the application, the digital watermarking technique can support different levels of robustness against changes made to the watermarked content. If digital watermarking is used for ownership identification, then the watermark has to be robust against any modifications. The watermarks should not get degraded or destroyed as a result of unintentional or malicious signal and geometric distortions like analog-to-digital conversion, digital-to-analog conversion, cropping, re-sampling, rotation, dithering, quantization, scaling and compression of the content. On the other hand, if digital watermarking is used for content authentication, the watermarks should be fragile, i.e., the watermarks should get destroyed whenever the content is modified.

The watermark must be difficult (hopefully impossible) to remove. If only partial knowledge is available (for example, the exact location of the watermark in an image is unknown), then attempts to remove or destroy a watermark should result in severe degradation in fidelity before the watermark is lost. In particular, the watermark should be robust in the following areas:

• Inseparability - After the digital content is embedded with watermark, separating the content from the watermark to retrieve the original content is not possible.

• Common Signal Processing - The watermark should still be retrievable even if common signal processing operations are applied to the data. These include, digital-to-analog and analog-to-digital conversion, re-sampling, re-quantization (including dithering and recompression), and common signal enhancements to image contrast and color, or audio bass and treble, for example.

• Common Geometric Distortions - Watermarks in image and video data should also be immune from geometric image operations such as rotation, translation, cropping and scaling.

• Subterfuge Attacks (Collusion and Forgery) - In addition, the watermark should be robust to collusion by multiple individuals who each possess a watermarked copy of the data. That is, the watermark should be robust to combining copies of the same data set to destroy the watermarks. Further, if a digital watermark is to be used in litigation, it must be impossible for colluders to combine their images to generate a different valid watermark with the intention of framing a third party.

• Universality - The same digital watermarking algorithm should apply to all three media under consideration. This is potentially helpful in the watermarking of multimedia products. Also, this feature is conducive to implementation of audio and image/video watermarking algorithms on common hardware.

• Unambiguousness - Retrieval of the watermark should unambiguously identify the owner. Furthermore, the accuracy of owner identification should degrade gracefully in the face of attack.

2.7.3 Security

The digital watermarking techniques prevent unauthorized users from detecting and modifying the watermark embedded in the cover signal. Watermark keys ensure that only authorized users are able to detect/modify the watermark. Finally, the watermark should withstand multiple watermarking to facilitate traitor tracing.

In general, a digital watermark should have several different properties. The most important are imperceptibility, robustness and security. Imperceptibility means that the watermarked data should be perceptually equivalent to the original, un-watermarked data. In some applications, the watermark may be perceptible as long as it is not annoying or obtrusive; however, many applications require that the watermark be imperceptible. Security means that unauthorized parties should not be able to detect or manipulate the watermark. Cryptographic methods are typically employed to make watermarks secure. Finally, robustness means that, given the watermarked data, one should not be able to make the watermark undetectable without also destroying the value or usefulness of the data.

Another characteristic of a watermarking scheme is whether or not the original data is available during detection. In some schemes [1], the watermark detector has access to the original data. Hence, interference from the original can presumably be eliminated. Blind schemes do not have the luxury of using the original during watermark detection. They typically apply some pre-processing to the received data to suppress interference from the original.

2.8 DESIGN CONSIDERATIONS AND REQUIREMENTS

Requirements and design of watermarking techniques are impacted by the different types of content in two major ways: imperceptibility and robustness requirements.

The first challenge is designing a watermark embedding algorithm which provides an imperceptible mark, that is, one which does not noticeably degrade the original host signal. Ideally, the marking algorithm should be adapted by using perceptual models appropriate for the different media types. The perceptual models used for representations of continuous tone images are not appropriate for text or graphics.

The other factor for designing watermarking schemes for multimedia is the type of degradations that the watermark is expected to survive and system requirements for media specific applications. For instance, it may be desirable for a still image watermarking technique to be able to survive JPEG compression and photocopying while for some video watermarking applications, it may be important to do watermark embedding and detection in real time on a compressed bit stream.

Moreover, the type of manipulations and the attacker expected computational power heavily depend on the application. Watermarking, like cryptography, also uses secret keys to map information to owners, although the way this mapping is actually performed considerably differs from what is done in cryptography, mainly because the watermarked object should keep its intelligibility. In most watermarking applications embedment of additional information is necessary. This information includes identifiers of the owner, recipient and/or distributor, transaction dates, serial numbers, etc. which play a crucial role in adding value to watermarking products.

2.9 STRUCTURE OF A WATERMARKING SYSTEM

Every watermarking system consists at least of two different parts: watermark embedding unit and watermark detection and extraction unit. Figure 2.3 shows an example of embedding unit for still images. The unmarked image is passed through a perceptual analysis block that determines how much a certain pixel can be altered so that the resulting watermarked image is indistinguishable from the original. This takes into account the human eye sensitivity to changes in flat areas and its relatively high tolerance to small changes in edges. After this so-called perceptual-mask has been computed, the information to be hidden is shaped by this mask and spread all over the original image. This spreading technique is similar to the interleaving used in other applications involving coding, such as compact disc storage, to prevent damage of the information caused by scratches or dust. In our case, the main reason for this spreading is to ensure that the hidden information survives cropping of the image. Moreover, the way this spreading is performed depends on the secret key, so it is difficult to recover the hidden information if one is not in possession of this key. Additional key-dependent uncertainty can be introduced in pixel amplitudes (recall that the perceptual mask imposes only an upper limit). Finally, watermark is added to the original image.



Figure 2.3 Watermark insertion unit





Figure 2.4 Original ‘Lena’ image and Perceptual Mask of the image

Figure 2.4 represents the perceptual mask that results after analyzing the image presented in Figure 2.4. Higher intensity (i.e., whiter) levels imply that higher perturbations can be made at those pixels without perceptible distortion. Thus, the higher capacity areas for hiding information correspond to edges. These masks are computed by using some known results on how the human eye works in the spatial domain. Different results are obtained when working on other domains, such as the DCT (Discrete Cosine Transform) or Wavelet transform. In fact, when working on the DCT coefficients domain one may take advantage of the relative independence between the maximum allowable perturbations at every coefficient. This is useful when dealing with the mask for watermarking purposes.



Figure 2.5 Watermark detection and extraction unit

Above, Figure 2.5 shows the typical configuration of a watermark detection and extraction unit. Watermark detection involves deciding whether a certain image has been watermarked with a given key. Note then that a watermark detector produces a binary output. Important considerations here are the probability of correct detection PD (i.e., the probability of correctly deciding that a watermark is present) and the probability of false alarm PF (i.e., the probability of incorrectly deciding that an image has been watermarked with a certain key). These two measures allow us to compare different watermarking schemes: One method will be superior if achieves a higher PD for a fixed PF. Note also that for a watermarking algorithm to be useful it must work with extremely low probabilities of false alarm.

Watermark detection is usually done by correlating the watermarked image with a locally generated version of the watermark at the receiver side. This correlation yields a high value when the watermark has been obtained with the proper key. It is possible to improve the performance of the detector by eliminating original image-induced noise with signal processing. It is worthy of remark that some authors, like Cox I.J. in [1], propose using the original image in the detection process.

Once the presence of the watermark has been correctly detected, it is possible to extract the hidden information. The procedure is also generally done by means of a cross-correlation but in this case, an independent decision has to be taken for every information bit with a sign slicer. In fact, I.J. Cox et al. [1] have also shown that this correlation structure has not been well-founded and significant improvements are achievable when image statistics are available. For instance, the widely-used DCT coefficients used in the JPEG and MPEG-2 standards are well approximated by generalized Gaussian probability density functions that yield a considerably different extraction scheme. Obviously, when extracting the information the most adequate parameter for comparison purposes is the probability of bit error Pb, identical to that used in digital communications. This is not surprising because watermarking creates a hidden (also called Stegnographic) channel on which information is conveyed.

**Chapter 3**

**Audio Watermarking Techniques**

3.1 INTRODUCTION

Originally, watermark referred to the invisible ink in official documents that one sees by holding a candle against the paper. Digital watermarking is the process of attaching the identity of a copyright owner to a digital audio work that is difficult to remove. The technique is analogous to a stain on a dress that is impossible to get out or the mark of floodwater upon a building after the tide has receded. The goal of digital audio watermarking is to give the artistic work owner the capability to prove the work is theirs.

There are many challenges to audio watermarking. The technical requirements are that the watermark does not add distortion, be audible in the signal, be robust against further audio processing, be not removable by attacks designed to separate the watermarking from the audio illegally and it must also be easily identifiable by a decoding mechanism.

Watermark Extraction, or the ability to detect and remove the watermark reliably is the most important point of a particular watermarking technique. Unreliable assessment of the copyright owner of course creates the question of why one would bother to hide data in the artistic work in the first place. This concept is analogous to the thief who never finds their buried treasure.

 Watermarks can have other useful purposes beyond copyright protection. For instance, one could include metadata and other information hidden in the original signal to be sent, bound with the audio. Metadata are textual descriptors, such as artist name, copyright, etcetera, for additional information on the artistic work. For example, I could send an Internet web site address (URL) in my compressed audio stream that tells the purchaser where to download the album art for this particular piece of music.

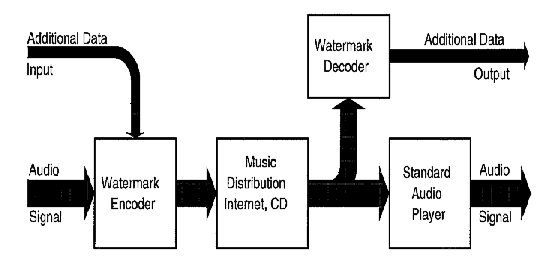


Figure 3.1 Overview of an audio watermarking system (Neubauer and Herre, 1998)

Any audio watermarking system has some basic components. An encoder to insert the audio watermark, and a decoder for extracting the audio watermark to identify the original owner. Furthermore, transactional watermarking, or unique watermarking per copy of an artistic work, must be done in real time, or a minimal, very rapidly (Webreference, 1998).

3.2 Requirements of Audio Watermarking

When first examining a problem, one needs to find the objectives that create a solution. This section discusses the requirements of audio watermarking in order to meet the goals of binding the artist owner, digitally, to their original artist work (Linnart, 1997).

Some other fundamental ideas are that detection of the watermark should not require the original cleartext artistic work to be detected. Cleartext means there is no watermark in the signal.

Table 3.1 Necessary conditions of audio watermarking algorithms

|  |
| --- |
| Inaudible |
| Survives Additional Audio Processing |
| Survives Bit stream and Post processing Audio Editing |
| Support for Multiple Applications |
| Bound to the Original Digital Audio |
| Easily Extracted by Detection Algorithm |
| Destroys Artistic work upon attack |

3.2.1 Inaudible

Artifacts, noise and other distortions should not be introduced by the watermark. A marketing statement on this attribute might be, "listen, you can't hear it!" as was used by Zhao (1998).

 3.2.2 Robustness to Current and Future Audio Compression Technologies

A watermark should survive perceptual audio coding; lossless coding and tandeming, for these technologies are deployed in a variety of digital audio applications to date.

Lossless audio compression is the compression of an audio signal with no loss of data. Audio watermarking will automatically survive this technique by the definition of lossless audio compression.

Perceptual audio coding is a lossy compression technology. The original audio signal is analyzed to see which parts, if removed, will not be perceived missing by the listener. The signal is then compressed and decompressed. The resulting decompressed signal is only the parts of the original audio work that are required by the listener to psychoacoustically reconstruct the original piece of audio to be perceived as the original. Therefore, the audio watermark should possess the same properties as the "important" parts of the original audio so it will not be removed.

Some applications also deploy the use of tandeming, or multiple encodings and decoding of the original audio signal. Audio watermarking should survive this process. If the audio watermark is removed during tandeming, then the audio quality should also be degraded.

Finally, there is a realm of audio and speech compression that is a quality level below the level of imperceptual difference between the original and the compressed-decompressed signal. This is in the realm of "AM quality" or "FM quality" or "magnetic tape" quality, i.e. it is "good enough". In this realm, audio watermarking might also be desirable. We all have seen bootlegged copies of famous artists rock concerts for example, that are certainly not of quality level as their desired released works, yet these performances are stolen and sold to adoring fans. Audio watermarking in this case could also assist in tracking these sorts of thefts and give the artist more control over what they wish to be released to the public and what they do not intend their public to experience.

3.2.3 Audio Signal Processing Robustness

Audio authoring tools, such as audio digital workstations perform pre and post processing of digital audio to insure the best audio quality for the application. Such pre and post processing includes, dynamic compression, limiters, equalization, A/D and D/A conversion, time scaling, pitch scaling, sample rate conversion, linear and non-linear filtering, mixing, addition of noise into the signal and finally noise reduction algorithms such as Dolby "C".

Also involved is the editing and cross fading of the audio work for such applications as commercials, movies, "dance version" of a song, etcetera.

All of these sorts of manipulations of digital audio data imply that the digital audio watermark must be extractable and readable after this sort of processing has been done.

3.2.4 Binding the Watermark to the Original Signal

The audio watermark must be inexplicably bound to the audio signal. This implies that the statistics of the audio watermark should possess similar statistics to the original audio signal. A thief could analyze this signal for "musical" statistics and "watermark" patterns then remove the hidden data. This also implies that the watermark should be imbedded in data that is critical for the final resulting audio signal. For example, one would not place the audio watermark in a file header that could easily be stripped out, yet one could still recover the actual audio within that file format.

 3.2.5 Overall Strategy of Watermark Placement

From the previous sections, we discover there is a certain strategy in the audio watermark placement, similar to hiding one's valuables in a hotel room. One should not place the watermark in place in the audio signal that is easy to detect and remove, such as in the LSB (least significant bit) of a 24bit resulting audio data type, or in one frequency bin of a Fast Fourier Transform. One should place the audio watermark where the perceptually critical aspects of the audio signal reside. The reason for this is that most audio processing applications and lossy compression technologies will leave only the significant portions of the audio signal intact. Otherwise these processing technologies will simply destroy the original audio signal. The goal is to bind the audio watermark, hence if the watermark is removed; the music piece is also destroyed beyond recognition to the original artistic work.

3.3 Watermarking Detection

There are two ways that a pirate can defeat a watermarking scheme. The first is to manipulate the audio signal to make all watermarks undetectable by any recovery mechanism. The second is to create a situation where the watermarking detection algorithm generates a false result that is equal to the probability of a true result (Boney, et al., 1996).

The detection of the watermarking signal is the most important aspect of the entire watermarking process. For if one cannot easily and reliably extract the actual data that was inserted in the original signal, it matters little what exotic techniques were used to perform this insertion. The watermark extraction will occur in the presence of jamming signals and the above real life harsh audio conditions.

 3.4 What Watermarking Should Not Be

From the previous criteria in section 3.1, there are certain watermarking techniques that should not be used. For example, any watermarking that always places the data into one frequency bin should not be utilized. The reason for this is, as a pirate, it is very easy for one to design a low pass filter, a high pass filter or a band reject filter that simply removes the watermark. Also, it is apparent that this sort of watermark automatically degrades the original audio signal.

Another poor technique is to place the watermark in moments of time where is does not matter that it is audible (so much). For example, in-between songs on a CD or in a place in the audio where the quality of the original work is so poor that the watermarked audio quality is as bad as the song. Again, this is an unacceptable technique. Simple audio editing tools could easily remove the watermark.

3.5 Techniques

Audio Watermarking is a newly established engineering field. Ongoing research and standardization efforts are currently in process. There have been identified four different areas in which audio watermarking have been attempted. Time domain based audio watermarking, audio watermarking in the frequency domain. Audio watermarking in the compressed domain and finally, combinations of the previous three techniques has research results presented. Some techniques combine audio compression technologies and others apply to clear text, or PCM data. PCM, or pulse code modulation is the standard representation of an audio signal in the digital realm. PCM are the bits stored on your musical CD.

3.5.1 Watermarking in the Time Domain

Manipulation of an audio signal in the time domain means that the work is done on the PCM data or time based data. The watermark is not inserted in the frequency domain. Frequency based techniques can be used to analyze the signal for proper placement of the digital audio watermark.

3.5.1.1 Changing the Least Significant Bits of Each Sample

In Bassia's technique (Bassia, 1998), a watermark is inserted by adding the watermark to the original audio signal x(i). The function f() is a noise floor calculation so that the watermark is not heard. The random sequence is represented by w(i) and is the watermark signal to be inserted.

http://www.musemagic.com/images/Image27.gif(1)

To extract the watermark, we first represent the sum of the final watermarked signal times the original watermark that was inserted. N is fixed and represented at least one second of sampled audio data.

http://www.musemagic.com/images/Image28.gif (2)

Then, by substitution, we replace y(i) in equation 2 with its original process. The result is equation 3. N is the same for all equations in this section.

http://www.musemagic.com/images/Image29.gif (3)

 By breaking up equation 3 into subsections, Bassia now manipulates the fact that w(i), the watermark signal, is basically gaussian white noise. Thus an equally biased audio signal times a gaussian white noise signal's mean is zero. Sometimes, the watermarked signal is not equally distributed random data. Therefore, one represents this portion of the watermarked signal by the sum from 1 to D w. The rest of the watermarked signal, represented by the first sum in equation 4, is assumed to have a mean of zero.

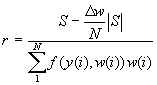
http://www.musemagic.com/images/Image30.gif (4)

 Then, since N is fixed, one can rewrite the second sum in equation for as equation (5)

http://www.musemagic.com/images/Image31.gif (5)

Firstly, if the signal is not watermarked, equation 4 reduces to equation 5. If the signal is watermarked the equation is now reduced to the last two sum terms of equation 4. Now, notice that the equation still relies on x(i), the original audio signal. This implies that we must possess the cleartext audio, which is something undesirable.

Here is the fudge. We approximate x(i) by replacing this signal in the last sum term of equation 4 with y(i). The detective value, denoted by r, thus becomes

 (6)

The normalized detection value does possess inaccuracies due to the approximation of the cleartext signal by the audio signal to be detected for watermarking.

3.5.1.3 Psychoacoustic Models to Perform Clear text Marking

Many watermarking techniques use psychoacoustic modeling to properly place the watermark. Often the watermark itself is represented by a shift keying technique, such as amplitude-shift keying or frequency-shift keying (Lathi, 1983). Usually the sinusoidally represented data is placed where the masking property of psychoacoustic models can be utilized.

The psychoacoustic model (Tiiki and Beex, 1996) for the MPEG-4 natural audio coder calculates the maximum distortion energy, or the masking threshold that still will not be perceived as such by the audio listener (Neubauer, et al., 1998). The basic steps to this calculation are described in more detail by Bosi, et al. (1996).

3.5.1.4 Echo Data Hiding

Echo data hiding is the idea of placing data very close, i.e. within microseconds, of the original signal in the time domain. (Bender, et al., 1996) Echo data hiding uses three parameters, initial amplitude, decay rate and offset. Figure 3.2 illustrates the elements of the technique. The delay plus offset is set at 1/1000 second. This equates to 1ms, which is below the lowest known delay of two signals the human ear can hear as separate. Then the delta is an additional time that is also below perceived signal separation. Thus both a zero and a one will not be perceived with the original signal. This is assuming that the amplitude of the added data is below the amplitude of the original signal.

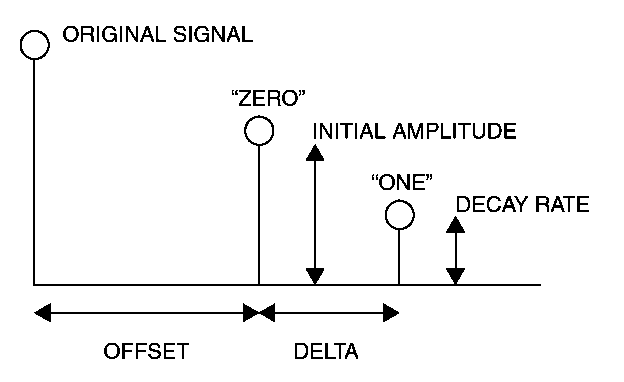


Figure 3.2 Parameters of echo data hiding (Bender, et. al, 1996)

If one draws a line connecting the original signal, the zero and the one, a decay envelope is created. This decay basically smears the original signal a bit, the line that one connected the dots with is the line perceived by the human ear as one signal, but this entire process is below the threshold of human perception, thus the signal is still perceived as the original one.

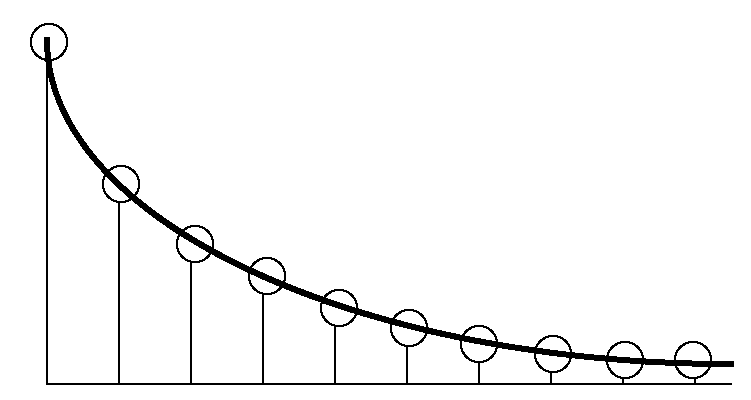


Figure 3.3 the auditory effect (Bender, et. al, 1996)

A sequence is encoded by representing a "zero" with offset and amplitude x and a "one" with delta + offset and amplitude y. To add more than 1 bit encoding to an entire signal, one simply breaks down the entire audio signal into small segments of length N. Then each audio segment is convolved with the data. A delta function, (which is basically what the "one" or "zero" is), convolved with the original signal produces a shift in the resulting output. Shift audio to where the "zero" or "one" is located on the time domain axis.

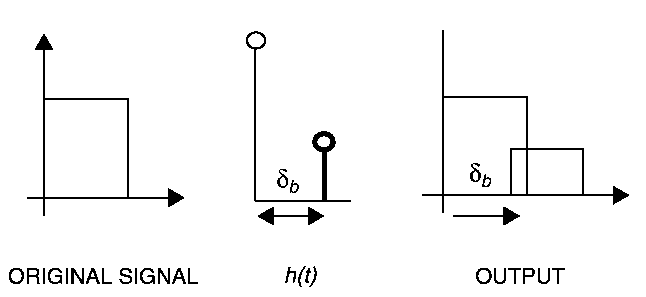


Figure 3.4 Small segments convolved with data (Bender, et. al, 1996)

Now the entire sequence segments of length N is put back together sequentially. This is the data sequence to be inserted into the original signal.

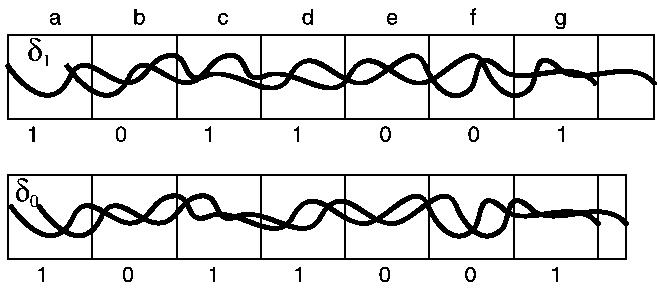


Figure 3.5 Reconstructed sequence delayed by "one", "zero" (Bender, et. al, 1996)

Next, create an original signal sequence, sifted initially by a "one", then another copy of the original audio sequence, shifted by a "zero". Multiply the "one" sequence by one, when a one is actually in the encoded data sequence for that segment of N and multiple the "zero" initially sifted sequence by the inverted original binary sequence to be inserted. Then add these two resulting signals together. The overall amplitude will be the same as the original signal. Also, this last mix step creates for a less noticeable distortion.

Decoding of the signal to extract the binary bit stream of data requires that one knows where delay of the one and the delay plus offset of the zero. The location is derived by taking the autocorrelation of the complex cepstrum of the original signal and the echoed version. The cepstrum is defined as the logarithm of the resulting output. The basic property here is that a power spectrum of signals with echo possesses an additive periodic component. Thus, the Fourier transform of the logarithm of the power spectrum exhibits a peak exactly where our beginning sequence is (Oppenheim and Schafer, 1989). Further details can be analyzed in (Bender, et al., 1996).

3.5.2 Watermarking in the Frequency Domain

For purposes of arbitrary categorization, this chapter defines Frequency Domain based watermarking on the insertion point of the watermark in Figure 3.6.

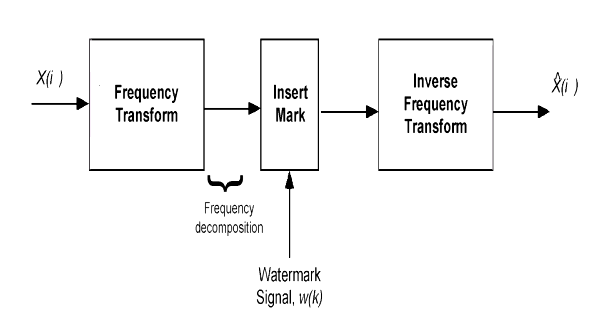


Figure 3.6 Frequency domain based watermarking (Cox, et al., 1995)

3.5.2.1 Phase Coding

Phase Coding can be an exceptional coding method. When the phase differential between the original signal and the modified one is kept small, it maintains close to the original un-watermarked audio's signal to perceived noise ratio (Bender, et al, 1996).

The idea behind phase coding is to hide the data by exchanging the original phase of the signal with the phase of the binary watermark plus the differential of the original audio phase. In a way, it is differential coding plus an binary phase offset. The reason for this is a randomly generated binary stream is a square wave. A square wave always possesses a phase of http://www.musemagic.com/images/Image38.gifor http://www.musemagic.com/images/Image39.gif.

The steps to perform phase coding are illustrated through the following figures. These figures were taken from (Bender, et al., 1996).

The first step is to break up the sound sequence into short segments as shown in figure 3.7. Each segment possesses fixed length of N.



Figure 3.7 Original signal and signal divided into segments (Bender, et al., 1996)

A discrete Fourier transform (Oppenheim and Schafer, 1983) is computed a on each segment of length N. For each ith element of phase data resulting from the DFT, up to N-1 values, compute the phrase differential between adjacent phase elements.

http://www.musemagic.com/images/Image41.gif

Figure 3.8 Magnitude and phase plots of DFT (Bender, et al., 1996)

The absolute phase of the watermark data signal is added to the differential, or D f of the last step. The result of this action is figure 3.8.

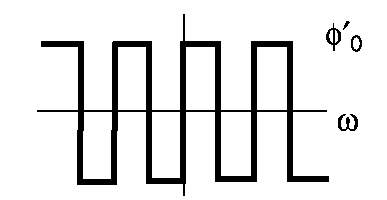


Figure 3.9 Data to be inserted, set this value as absolute phase (Bender, et al., 1996)

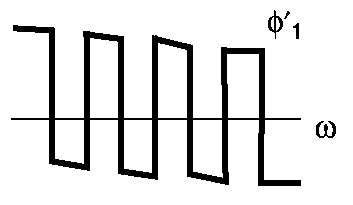


Figure 3.10 Adding watermark absolute phase to D f of original (Bender, et al., 1996)

Next, perform the inverse DFT on each segment with the original magnitude and the new modified phase value. The results should look like figure 3.11.

http://www.musemagic.com/images/Image44.gif

Figure 3.11 Resulting watermarking signal with phase coding (Bender, et al., 1996)

Phase coding relies on the fact that human are much more sensitive to relative phase differences in an audio signal versus requiring an absolute phrase reference. Decoding is performed by synchronization. The initial phase value is detected as a zero or one.

3.5.2.2 Spread Spectrum

Spread spectrum is the technique of spreading data over the frequency spectrum of the original signal. There are two types of categories for spread spectrum, frequency hopping and direct sequence. Both types of spread spectrum require that the transmitter and receiver are synchronized. Frequency hopping is basically moving about the spectra in a pseudo random pattern. The frequency spectra is divided into bands and then the signal to be hidden is moved from one band to another in rapid succession. Direction sequence, is the multiplication of the original audio signal by a binary sequence in the encoder. Direct Sequence Spread Spectrum is often referred to as DSSS.

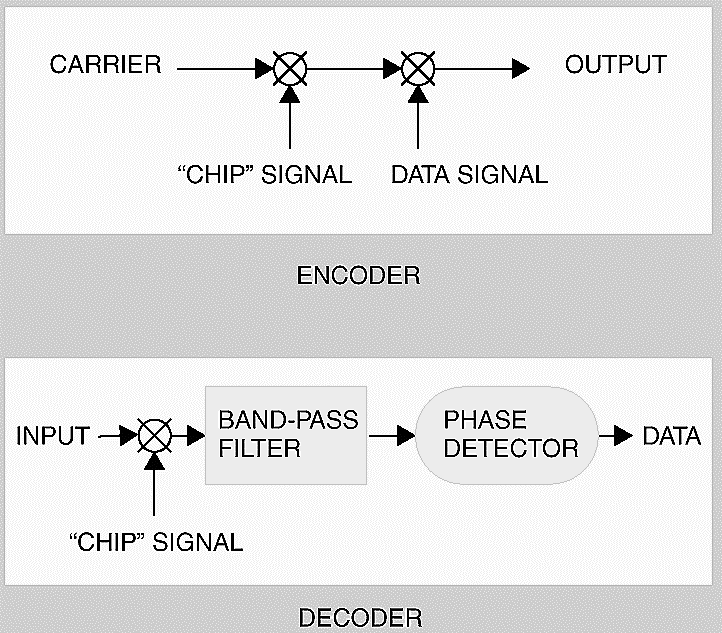


Figure 3.12 Basic DSSS spread spectrum DSSS (Bender, et al., 1996)

In figure 3.12, the chip is the name given to the pseudorandom sequence, which is modulated by the carrier rate. The chip rate has its own sampling frequency. Pseudorandom noise basically has similar properties as white noise, i.e. flat spectra across the entire frequency, Gaussian distribution and zero mean (Ziemer and Tranter, 1990). Another interesting property of pseudo random sequences is that the auto correlation of the PNS is at peak when delay is zero and minimal, or approximately zero for all other delays. For refreshment, the autocorrelation function is defined as:

http://www.musemagic.com/images/Image46.gif (7)

The chip rate must be many times greater than the data rate. This property is illustrated in equation 8. T represents the period of each signal.

http://www.musemagic.com/images/Image47.gif (8)

Figure 3.13 illustrates the DSSS encoding technique. The chip is basically the key needed by both the encoder and decoder to modulate the data sequence. The binary code is multiplied by the carrier wave and also by the chip or pseudorandom binary sequence. In this case, the carrier wave represents the frequency band that the data is being spread over. This result is then attenuated and added to the original audio signal. The effect is similar to adding white noise, due to the multiplication by the pseudorandom sequence to the data.

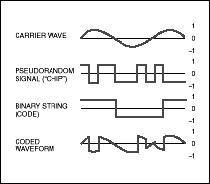


Figure 3.13 Spread spectrum process (Bender et al., 1996)

To decode the hidden data, one must phase lock loop to the chip frequency and also know the start of the chip, or pseudo-random noise sequence. A PLL, or phase lock loop detects the phase of the incoming signal and locks upon the signal (Ziemer and Tranter, 1990). The data rate must also be know, in order to synchronize up to the data of the received signal.

By its nature this technique is adding noise to the signal, thus spread spectrum may be at odds with advances in compression technology at low bit rates.

3.5.3 Watermarking in the Compressed Domain

Compressed domain watermarking means manipulation of the bit stream. This implies that one does not change the original bit stream syntax. The example given by (Lacy, et al., 1998) is to apply a data envelope to an MPEG-2 Advanced Audio Coding (AAC) bit stream. There are other techniques to hide data in the bit stream and also MIDI data hiding techniques. The main point is to make decoding the bit stream dependent on the watermark existing in it.

3.6 Combinations of Watermarking Techniques

3.6.1 Psychoacoustic Models and Spread Spectrum

Neubauer and Herre (1998) approached the audio watermarking solution by also utilizing a multi-dimensional approach. This work is an extension of the previous work of Boney (1996). Figure 3.14 illustrates the encoding process. There are three main sections, modulation, signal conditioning and the input/output audio process.

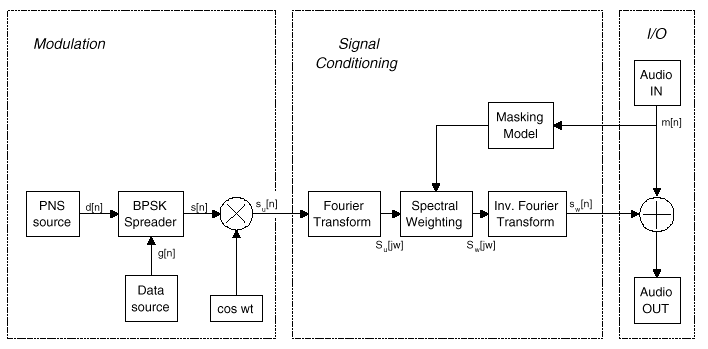


Figure 3.14 Psychoacoustic model and DSSS encoding (Neubauer, et al., 1998)

The masking model used here is the same process described in section 3.1.2. The main idea is to determine where one can place the data, now in the form of a modulated signal, in the audio data. Normally, one must note that it is possible for the data to be removed by the psychoacoustic model. Certainly data carrier at http://www.musemagic.com/images/Image50.gifis susceptible to a psychoacoustic model.

The spectral weighting function is designed to scale the frequency domain representation of the modulated watermarked signal so that it will always be hidden below the masking threshold of the audio. For each critical band the energy of the spread spectrum signal is weighted to be the energy computed by the psychoacoustic model. This does imply that the watermark could be lost during periods of silence or of extreme low energy, but even this is not that much of a problem, as we shall see in the decoder description.

Modulation is very similar to the spread spectrum technique described in section

3.6.2 There are five main components in this diagram

PNS source - pseudo noise sequence

Data source - watermark data to be inserted into the audio signal

BPSK Spreader - binary phase shift keying (Proakis, 1995)

Multiplication process

Carrier wave - cos(wt)

 This technique is called DSSS-BPSK modulation (Proakis, 1995). In this technique the BPSK spreader is more complex than the spread spectrum technique of (Bender, 1996). In Bender's technique the BPSK spreader is the second multiplication in equation 9. Neubauer (1998) implies that the BPSK spreader breaks down to a multiply. The reason for this is the data and the PNS possess opposite values, i.e. -1, +1, for all time. The pseudo noise sequence is a bipolar, N length maximum sequence or m-sequence. Similarly to the previous PNS property discussion in the spread spectrum section, an m-sequence has this autocorrelation function:

http://www.musemagic.com/images/Image51.gif (9)

The decoder will use this property of the m-sequence to detect the watermark. The matched filter, seen in figure 3.15, inverts the data sequence so it is not reversed, i.e. 1000 because 0001. This sequence now becomes the matched filter's coefficients. Note that by multiplying the original sequence by the inverse of that sequence, one is computing the autocorrelation when j=0. Thus, when the gate closes the threshold decision is true upon receiving value N. The synchronization unit is there to count the N length of the sequence. Since the energy of the watermarked signal has been scaled, this aspect of the algorithm is particularly important to insure that proper sequencing occurs.

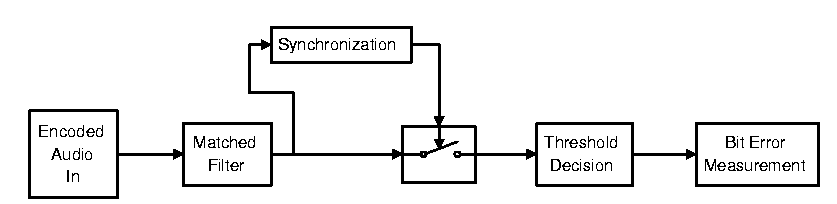


Figure 3.15 Decoder (Neubauer, et al., 1998)

3.6.3 Frequency Domain Shaping and Time Domain Weighting

Laurence Boney, et al., introduces a technique by utilizing the psychoacoustic masking ability of the MPEG Psychoacoustic Model (MPEG, 1993). Boney creates a Frequency Domain Based Masking Filter on the PN Sequence representing the data to be inserted and then a time domain energy based weighting to insure that the inserted watermark is above the MPEG audio compression algorithm quantization level.

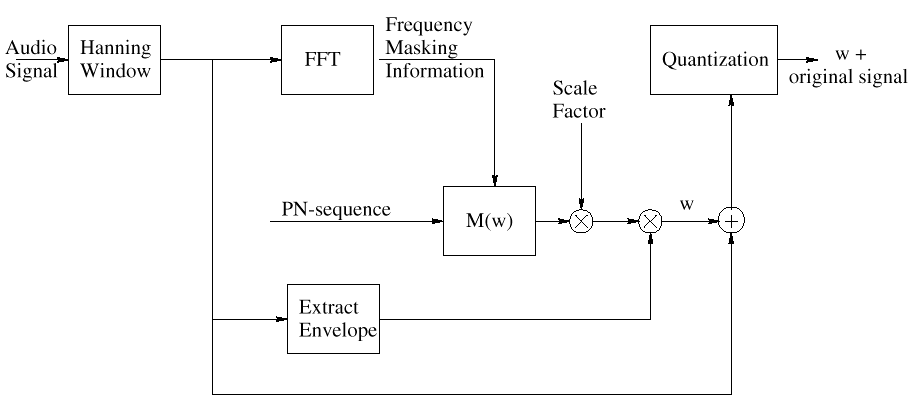


Figure 3.16 Generator for masking filter data insertion (Boney, et. al, 1996)

The window length used in is 512 samples, weighted with a Hanning window. Overlapped transform of 50% is used. The masking threshold is approximated with a 10th order all poll filter M(w). This technique is performed using a least squares approximation technique (MPEG, 1993). Then the PN-sequence, S(w), is filtered with the masking filter M(w) to insure that the watermark PN-sequence spectrum is below the masking threshold for this particular block of audio being processed. As one can see from figure 3.14, the watermark is then additionally weighted with a scale factor, multiplied by the weighted Hanning Window and finally inserted into the audio spectrum before proceeding to the quantization phase of the encoding process.

3.6.4 Psychoacoustic Masking and Bit stream Watermarking

ATT Research Labs presented a technique that incorporated bit stream watermarking with psychoacoustic masking. This technique is only described in conjunction with MPEG2-AAC audio. The tools listed that are used are the psychoacoustic model, rate control, quantization and noiseless, or huffman coding blocks. In this system, the audio watermarking procedure is to choose scale factor bands to be marked via the perceptual model from the MPEG-2 audio. These scale factor bands, from the spectral lines of AAC, are then sent to a quantizer step size and Huffman table. Lacy implies that the Huffman table representing null Huffman codes, or noise, should not be watermarked. This implies that one is hiding the data via some sort of sinusoidal masking concept. This particular table is not transmitted, as there is no spectral data to send.

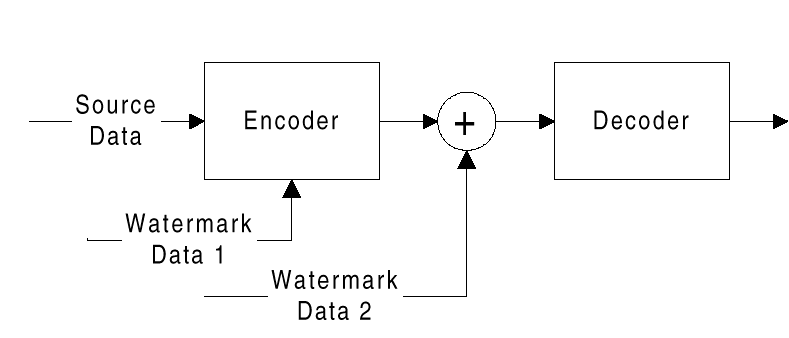


Figure 3.17 Block diagram of ATT's combined technique (Lacy, et al., 1998)

The original quantizers, or scale factors of MPEG2-AAC, are divided by a set of multipliers. An offset vector, representing the watermark data, is added to the newly scaled quantizers. There are three techniques or methods to perform this operation as presented by Lacy, et al. (1998).

In order to find the beginning of the sequence, Lacy used the frame boundary of the AAC bit stream. The initial scale factors are modified in the LSB in order to contain a series of synchronization codes.

One of the more interesting aspects of this placement is the fact that random flipping of a scale factor LSB will produce artifacts. This is unlike the situation in manipulating the LSB of a time domain based signal. Due to the conservation of bits, i.e. compression, all data matters.

The applications of watermarking are vast. Hence, watermarks often perform different functions based upon the application. These three functions are: identification of the original of the content, tracing illegally distributed copies of the content and disabling unauthorized access to the content (Lacy, et al., 1997).

In this section, the advantages and disadvantages of each technique are listed in table form. Audio quality issues are also discussed. No one algorithm, to date, has deemed to be perfected. Hence, the quest for better algorithms and more secure networks is ongoing.

3.7 Time Domain

The issue with this technique is computational complexity and I must know the actual watermark. If I do not know the watermark, I must go through 2x combinations of bits, where x is the frame size, to actually identify the watermark. Thus, for transactional watermarking, where the decoder is identifying the client, this technique is not applicable.

Table 3.2 Advantages and disadvantages of least significant bit insertion

|  |  |
| --- | --- |
| Too many Assumptions | Disadvantage |
| Highly Susceptible to Attack | Disadvantage |
| Requires no Decoder | Advantage |
| Potentially reversed Engineered by Player for Removal | Disadvantage |

Psychoacoustic models and data placement fall short when one then compresses the audio at a later date with some perceptual coder. The psychoacoustic model in the compression algorithm performs the same task as the watermark encoder. It seeks redundant portions of the audio signal that will not be perceived. Since one of these redundancies happens to be your carefully placed watermark, It is now removed.

Table 3.3 Advantages and disadvantages of psychoacoustic models and cleartext

|  |  |
| --- | --- |
| Too Complex for Transactional | Disadvantage |
| Susceptible to Advances in Perceptual Models | Disadvantage |
| Survives all Processed Generations | Advantage |
| Potentially reversed Engineered by Player for Removal | Disadvantage |

The most difficult aspect of echo hiding is the computation of the complex cepstrum space to decode the watermark. It does not require the original watermark to be detected. It is also susceptible to psychoacoustic models removing the watermark.

Table 3.4 Advantages and disadvantages of echo data hiding

|  |  |
| --- | --- |
| Too complex for transactional | Disadvantage |
| Watermark detection difficult | Disadvantage |
| Not robust for compression | Disadvantage |
| Not robust for resonance audio processing | Disadvantage |

3.8 Frequency Domain

The main difficulty with spread spectrum is that the noise added eventually with be at odds with any perceptual coder. Thus, the watermark to be inaudible, under all bitrates is the challenge of this technique. The synchronization methods do require that the PNS data rate is known, but do not require the actual watermark.

Table 3.5 Advantages and disadvantages of spread spectrum

|  |  |
| --- | --- |
| Synchronization | Disadvantage |
| Audible | Disadvantage |
| Bound to audio | Advantage |
| Reliable Detection | Advantage |

Phase coding is also a candidate to run at odds with lossy compression technology and further audio processing. There are many audio processes that manipulate the phase of the audio signal. This may interfere with the watermark being detected.

Table 3.6 Advantages and disadvantages of phase coding

|  |  |
| --- | --- |
| Synchronization | Disadvantage |
| Data Interval | Disadvantage |
| Watermark not needed | Advantage |
| Low Distortion | Advantage |

3.9 Compressed Domain

Hiding information within a bitstream is highly desirable. There is little processing involved and the information is easy to parse unfortunately, bitstreams are more easily hacked that the original audio by computer hackers. Certainly one can be incredibly clever and create data dependencies with the bitstream and the watermark, but unfortunately, hackers are also usually quite clever. The watermark is also not audible.

Table 3.7 Advantages and disadvantages of compressed domain

|  |  |
| --- | --- |
| Cannot Survive D/A conversion | Disadvantage |
| Not Robust against Attack | Disadvantage |
| Low Complexity | Advantage |
| Supports All Bitrates | Advantage |

3.10 Combined Techniques

Most of these techniques are the ones most recently developed. As research has progressed, it has become apparent in the perceptual audio coding world that binding the watermark to the actual encoding process is a much better idea to insure the watermark is not audible. Bitrates are now in the 8kbps range for audio coding, hence there is very little redundant information for which to hide the watermark behind. By combining techniques, one has a better assurance that the watermark will not be audible under all bitrates of a compression technology.

Table 3.8 Frequency and time domain shaping and weighting

|  |  |
| --- | --- |
| Robust to Processing | Advantage |
| Access to Original PN Sequence | Disadvantage |
| Synchronization | Disadvantage |
| Reliable Detection | Advantage |

This technique has low computational complexity, thus is ideal for transactional, or per sale, watermarking. It also does not degrade the audio quality. One issue will be it’s susceptibility to attack and also the watermark is lost upon D/A conversion and also after the bitstream is decoded.

Table 3.9 Psychoacoustic modeling and bitstream marking

|  |  |
| --- | --- |
| Synchronization | Disadvantage |
| Computational Complexity Low | Advantage |
| Bitrate of Encoded Audio | Disadvantage |
| Reliable Detection | Advantage |

The use of a watermark algorithm bound to the compression technology is interesting. It stops the "at odds" problem of adding data to a process designed to extract redundant audio information. Yet, experimental results reveal that this technique still creates some noticeable audio distortion. The technique is robust against attack and also survives D/A conversion and other signal processing techniques.

Table 3.10 Psychoacoustic modeling and spread spectrum

|  |  |
| --- | --- |
| Synchronization | Disadvantage |
| Computational Complexity Low | Advantage |
| Bitrate of Encoded Audio | Disadvantage |
| Reliable Detection | Advantage |

3.11 APPLICATIONS

Currently, there are many companies promoting their particular audio watermarking solution. To date derived from various corporate web sites; the author ascertained which techniques each company was using. Table 3.11 lists these overall approaches.

Table 3.11 some corporate techniques

|  |  |
| --- | --- |
| Solana Technology | Frequency Hopping Spread Spectrum |
| Aris Technologies | Direct Sequence Phase Keying |
| FhG | Spread Spectrum and Psychoacoustic Masking |
| Giovanni | Perceptual Convolution and FFT Analysis |
| Arbitron | Spread Spectrum and Psychoacoustic Masking |

The ISO MPEG-4 group has rejected standardization of audio watermarking in the multimedia standard. If one standardizes watermarking, one must also publish, in detail, how watermarking is done, in order for all interested parties to build such a device. This fact defeats the purpose of watermarking, for it gives the solution to all who can read the standard and run the verification model code provided by the MPEG committee. Also, one may not necessarily desire one particular watermarking method in certain applications. Therefore, convergence on one methodology will most likely occur in product consortiums, such as the DVD audio group.

Currently, the Copy Protection Technical Working Group is requesting industry to meet contradictory goals. Ultra low complexity in watermark detection, yet the best system industry can provide. The Data Hiding SubGroup, a Cross-industry body, has listed 13 goals deemed essential to the protection scheme (Yoshida, 1998b). These goals are listed in table 3.12.

Table 3.12 DHSG digital-video watermarking technology goals

|  |
| --- |
| Transparency |
| Low-cost digital Detection |
| Digital-detection domain |
| Generation copy control for one copy |
| Low false-positive detection |
| Reliable detection |
| Watermark will survive normal video processing in consumer use |
| Licensable under reasonable terms |
| Export/import status |
| Technical maturity |
| Data payload- watermark system should carry at least 8 bits of information |
| Minimum impact on content preparation |
| Data rate-minimum 11.08Mbytes/s to 25Mbytes/s to 270 Mbytes for video |

Audio Watermarking is now coming to be a critical strategy technology to insure theft does not occur of artistic works. The business model that relied on the failings of technology to inadequately reproduce copies of original works is no longer valid. Some watermarking techniques have been presented. Yet, the application and widespread use of digital audio watermarking is still under standardization and development. Certainly for the music and film industry to continue in the manner of allowing artists to receive payment for their contributions to humanity, digital watermarking must be incorporated into existing products. Else, we return to the days where artists are not properly acknowledged and supported for their enhancement of the human condition.

**Chapter 4**

**FUNDAMENTAL THEORY OF PROPOSED ALGORITHM**

4.1 Types of Audio formats

An audio file format is a [file format](http://en.wikipedia.org/wiki/File_format) for storing [audio](http://en.wikipedia.org/wiki/Sound) data on a [computer](http://en.wikipedia.org/wiki/Computer) system. It can be a raw [bitstream](http://en.wikipedia.org/wiki/Bitstream), but it is usually a [container format](http://en.wikipedia.org/wiki/Container_format_%28digital%29) or an audio data format with defined storage layer.

The general approach towards storing digital audio is to [sample](http://en.wikipedia.org/wiki/Sample_%28signal%29) the audio voltage which, on playback, would correspond to a certain level of signal in an individual channel with a certain [resolution](http://en.wikipedia.org/wiki/Audio_bit_depth)—the number of bits per sample—in regular intervals (forming the [sample rate](http://en.wikipedia.org/wiki/Sample_rate)). This data can then be stored uncompressed, or [compressed](http://en.wikipedia.org/wiki/Audio_compression_%28data%29) to reduce the file size.

It is important to distinguish between a [file format](http://en.wikipedia.org/wiki/File_format) and a [codec](http://en.wikipedia.org/wiki/Codec). A codec performs the encoding and decoding of the raw audio data while the data itself is stored in a file with a specific audio file format. Most of the publicly documented audio file formats can be created with one of two or more encoders or codecs. Although most audio file formats support only one type of audio data (created with an [audio coder](http://en.wikipedia.org/wiki/Audio_codec)), a multimedia container format (as [MKV](http://en.wikipedia.org/wiki/MKV) or [AVI](http://en.wikipedia.org/wiki/Audio_Video_Interleave)) may support multiple types of audio and video data.

There are three major groups of audio file formats:

* Uncompressed audio formats, such as [WAV](http://en.wikipedia.org/wiki/WAV), [AIFF](http://en.wikipedia.org/wiki/AIFF), [AU](http://en.wikipedia.org/wiki/Au_file_format) or [raw](http://en.wikipedia.org/wiki/Raw_audio_format) header-less [PCM](http://en.wikipedia.org/wiki/PCM);
* formats with [lossless](http://en.wikipedia.org/wiki/Lossless_data_compression) compression, such as [FLAC](http://en.wikipedia.org/wiki/Free_Lossless_Audio_Codec), [Monkey's Audio](http://en.wikipedia.org/wiki/Monkey%27s_Audio) ([filename extension](http://en.wikipedia.org/wiki/Filename_extension) APE), [WavPack](http://en.wikipedia.org/wiki/WavPack) ([filename extension](http://en.wikipedia.org/wiki/Filename_extension) WV), [Shorten](http://en.wikipedia.org/wiki/SHN), [TTA](http://en.wikipedia.org/wiki/TTA_%28codec%29), [ATRAC](http://en.wikipedia.org/wiki/ATRAC) Advanced Lossless, [Apple Lossless](http://en.wikipedia.org/wiki/Apple_Lossless), [MPEG-4 SLS](http://en.wikipedia.org/wiki/MPEG-4_SLS), [MPEG-4 ALS](http://en.wikipedia.org/wiki/MPEG-4_ALS), [MPEG-4 DST](http://en.wikipedia.org/wiki/MPEG-4_DST), [Windows Media Audio Lossless (WMA Lossless)](http://en.wikipedia.org/wiki/Windows_Media_Audio#Windows_Media_Audio_Lossless).
* formats with [lossy](http://en.wikipedia.org/wiki/Lossy_data_compression) compression, such as [MP3](http://en.wikipedia.org/wiki/MP3), [Vorbis](http://en.wikipedia.org/wiki/Vorbis), [Musepack](http://en.wikipedia.org/wiki/Musepack), [AAC](http://en.wikipedia.org/wiki/Advanced_Audio_Coding), [ATRAC](http://en.wikipedia.org/wiki/ATRAC) and lossy [Windows Media Audio](http://en.wikipedia.org/wiki/Windows_Media_Audio) (WMA).

4.1.1 Uncompressed audio formats

There is one major uncompressed audio format, [PCM](http://en.wikipedia.org/wiki/PCM), which is usually stored as a .wav on [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) or as .aiff on [Mac OS](http://en.wikipedia.org/wiki/Mac_OS). WAV and AIFF are flexible file formats designed to store more or less any combination of sampling rates or bitrates. This makes them suitable file formats for storing and archiving an original recording. There is another uncompressed audio format which is .cda (Audio CD Track) .cda is from a music cd and is 0% compressed.

The AIFF format is based on the [IFF](http://en.wikipedia.org/wiki/Interchange_File_Format) format. The WAV format is based on the [RIFF](http://en.wikipedia.org/wiki/RIFF_%28File_format%29) file format, which is similar to the IFF format.

[BWF](http://en.wikipedia.org/wiki/Broadcast_Wave_Format) (Broadcast Wave Format) is a standard audio format created by the [European Broadcasting Union](http://en.wikipedia.org/wiki/European_Broadcasting_Union) as a successor to WAV. BWF allows [metadata](http://en.wikipedia.org/wiki/Metadata) to be stored in the file. See [European Broadcasting Union: Specification of the Broadcast Wave Format](http://en.wikipedia.org/wiki/Broadcast_Wave_Format) (EBU Technical document 3285, July 1997). This is the primary recording format used in many professional audio workstations in the television and film industry. BWF files include a standardized [Timestamp](http://en.wikipedia.org/wiki/Timestamp) reference which allows for easy synchronization with a separate picture element. Stand-alone, file based, multi-track recorders from Sound Devices[[1]](http://en.wikipedia.org/wiki/Audio_file_format#cite_note-0), Zaxcom[[2]](http://en.wikipedia.org/wiki/Audio_file_format#cite_note-1), HHB USA[[3]](http://en.wikipedia.org/wiki/Audio_file_format#cite_note-2), [Fostex](http://en.wikipedia.org/wiki/Fostex), and Aaton[[4]](http://en.wikipedia.org/wiki/Audio_file_format#cite_note-3) all use BWF as their preferred format.

4.1.2 Lossless compressed audio formats

A lossless compressed format requires more processing time than an uncompressed format but is more efficient in space usage.

Uncompressed audio formats encode both sound and silence with the same number of bits per unit of time. Encoding an uncompressed minute of absolute silence produces a file of the same size as encoding an uncompressed minute of symphonic orchestra music. In a lossless compressed format, however, the music would occupy a marginally smaller file and the silence takes up almost no space at all.

Lossless compression formats (such as the most widespread[[5]](http://en.wikipedia.org/wiki/Audio_file_format#cite_note-4) [FLAC](http://en.wikipedia.org/wiki/FLAC), [WavPack](http://en.wikipedia.org/wiki/WavPack), [Monkey's Audio](http://en.wikipedia.org/wiki/Monkey%27s_Audio), [ALAC](http://en.wikipedia.org/wiki/Apple_Lossless)/Apple Lossless) provide a compression ratio of about 2:1. Development in lossless compression formats aims to reduce processing time while maintaining a good compression ratio.

4.1.3 Free and open file formats

* [wav](http://en.wikipedia.org/wiki/Wav) – standard audio file container format used mainly in [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) PCs. Commonly used for storing uncompressed ([PCM](http://en.wikipedia.org/wiki/PCM)), CD-quality sound files, which means that they can be large in size—around 10 MB per minute. Wave files can also contain data encoded with a variety of (lossy) codecs to reduce the file size (for example the GSM or mp3 codecs). Wav files use a [RIFF](http://en.wikipedia.org/wiki/Resource_Interchange_File_Format) structure.
* [ogg](http://en.wikipedia.org/wiki/Ogg) – a free, open source container format supporting a variety of codecs, the most popular of which is the audio codec Vorbis. Vorbis offers compression similar to MP3 but is less popular.
* [mpc](http://en.wikipedia.org/wiki/Musepack) - Musepack or MPC (formerly known as MPEGplus, MPEG+ or MP+) is an open source lossy audio codec, specifically optimized for [transparent](http://en.wikipedia.org/wiki/Transparency_%28data_compression%29) compression of stereo audio at bitrates of 160–180 kbit/s.
* [flac](http://en.wikipedia.org/wiki/Flac) – Free Lossless Audio Codec, a lossless compression codec.
* [aiff](http://en.wikipedia.org/wiki/Aiff) – the standard audio file format used by Apple. It is like a wav file for the [Mac](http://en.wikipedia.org/wiki/Macintosh).
* [raw](http://en.wikipedia.org/wiki/Raw_audio_format) – a raw file can contain audio in any codec but is usually used with PCM audio data. It is rarely used except for technical tests.
* [au](http://en.wikipedia.org/wiki/Au_file_format) – the standard audio file format used by [Sun](http://en.wikipedia.org/wiki/Sun_Microsystems), [Unix](http://en.wikipedia.org/wiki/Unix) and [Java](http://en.wikipedia.org/wiki/Java_%28Sun%29). The audio in au files can be [PCM](http://en.wikipedia.org/wiki/Pulse-code_modulation) or compressed with the [μ-law](http://en.wikipedia.org/wiki/M-law), [a-law](http://en.wikipedia.org/wiki/A-law) or [G729](http://en.wikipedia.org/wiki/G729) codecs.

4.1.4 Open file formats

* [gsm](http://en.wikipedia.org/wiki/GSM-FR) – designed for telephony use in Europe, gsm is a very practical format for telephone quality voice. It makes a good compromise between file size and quality. Note that wav files can also be encoded with the gsm codec.
* [dct](http://en.wikipedia.org/wiki/Dct) – A variable codec format designed for dictation. It has dictation header information and can be encrypted (often required by medical confidentiality laws).
* [vox](http://en.wikipedia.org/wiki/VOX_%28file_format%29) – the vox format most commonly uses the Dialogic [ADPCM](http://en.wikipedia.org/wiki/ADPCM) (Adaptive Differential Pulse Code Modulation) codec. Similar to other ADPCM formats, it compresses to 4-bits. Vox format files are similar to wave files except that the vox files contain no information about the file itself so the codec sample rate and number of channels must first be specified in order to play a vox file.
* [aac](http://en.wikipedia.org/wiki/Advanced_Audio_Coding) – the Advanced Audio Coding format is based on the [MPEG2](http://en.wikipedia.org/wiki/MPEG2) and [MPEG4](http://en.wikipedia.org/wiki/MPEG4) standards. aac files are usually [ADTS](http://en.wikipedia.org/wiki/ADTS) or [ADIF](http://en.wikipedia.org/wiki/ADIF) containers.
* [mp4](http://en.wikipedia.org/wiki/Mp4)/[m4a](http://en.wikipedia.org/wiki/M4a) – MPEG-4 audio most often AAC but sometimes MP2/MP3, MPEG-4 SLS, CELP, HVXC and other audio object types defined in [MPEG-4 Audio](http://en.wikipedia.org/wiki/MPEG-4_Audio)
* [mmf](http://en.wikipedia.org/w/index.php?title=Mmf&action=edit&redlink=1) - a Samsung audio format that is used in ringtones.

4.2 Proprietary formats

* [mp3](http://en.wikipedia.org/wiki/Mp3) – MPEG Layer-3 format is the most popular format for downloading and storing music. By eliminating portions of the audio file that are essentially inaudible, mp3 files are compressed to roughly one-tenth the size of an equivalent PCM file while maintaining good audio quality.
* [wma](http://en.wikipedia.org/wiki/Windows_Media_Audio) – the popular Windows Media Audio format owned by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). Designed with [Digital Rights Management](http://en.wikipedia.org/wiki/Digital_Rights_Management) (DRM) abilities for copy protection.
* [atrac](http://en.wikipedia.org/wiki/Atrac) (.wav) – the older style Sony ATRAC format. It always has a .wav file extension. To open these files simply install the ATRAC3 drivers.
* [ra](http://en.wikipedia.org/wiki/RealAudio) – a [Real Audio](http://en.wikipedia.org/wiki/Real_Audio) format designed for streaming audio over the Internet. The .ra format allows files to be stored in a self-contained fashion on a computer, with all of the audio data contained inside the file itself.
* ram – a text file that contains a link to the Internet address where the Real Audio file is stored. The .ram file contains no audio data itself.
* [dss](http://en.wikipedia.org/w/index.php?title=Dss&action=edit&redlink=1) – Digital Speech Standard files are an [Olympus](http://en.wikipedia.org/wiki/Olympus_Corporation) proprietary format. It is a fairly old and poor codec. Prefer gsm or mp3 where the recorder allows. It allows additional data to be held in the file header.
* [msv](http://en.wikipedia.org/w/index.php?title=Msv&action=edit&redlink=1) – a [Sony](http://en.wikipedia.org/wiki/Sony) proprietary format for Memory Stick compressed voice files.
* [dvf](http://en.wikipedia.org/w/index.php?title=Dvf&action=edit&redlink=1) – a Sony proprietary format for compressed voice files; commonly used by Sony dictation recorders.
* IVS – A proprietary version with [Digital Rights Management](http://en.wikipedia.org/wiki/Digital_Rights_Management) developed by 3D Solar UK Ltd for use in music downloaded from their [Tronme](http://en.wikipedia.org/w/index.php?title=Tronme&action=edit&redlink=1) Music Store and interactive music and video player.
* [m4p](http://en.wikipedia.org/wiki/M4p) – A proprietary version of AAC in MP4 with [Digital Rights Management](http://en.wikipedia.org/wiki/Digital_Rights_Management) developed by Apple for use in music downloaded from their [iTunes](http://en.wikipedia.org/wiki/ITunes) Music Store.
* [iklax](http://en.wikipedia.org/wiki/Iklax) – An iKlax Media proprietary format, the iKlax format is a multi-track digital audio format allowing various actions on musical data, for instance on mixing and volumes arrangements.
* [mxp4](http://en.wikipedia.org/w/index.php?title=Mxp4&action=edit&redlink=1) – a Musinaut proprietary format allowing play of different versions (or skins) of the same song. It allows various interactivity scenarios between the artist and the end user.
* [3gp](http://en.wikipedia.org/wiki/3gp) - multimedia container format can contain proprietary formats as [AMR](http://en.wikipedia.org/wiki/AMR), [AMR-WB](http://en.wikipedia.org/wiki/AMR-WB) or [AMR-WB+](http://en.wikipedia.org/wiki/AMR-WB%2B), but also some open formats
* [amr](http://en.wikipedia.org/wiki/Adaptive_Multi-Rate) - AMR-NB audio, used primarily for speech
* [awb](http://en.wikipedia.org/wiki/Adaptive_Multi-Rate_Wideband) - AMR-WB audio, used primarily for speech

4.3 Synchronization Code (Barkers Code)

Synchronization is one of the key issues of audio watermarking and losing synchronization cause false detection. Time scale and frequency scale modification can cause lost of synchronization. So there is need of an exact synchronization algorithm based on robust synchronization code.

The other problem with synchronization code is false synchronization. Generally, false synchronization should be avoided during selecting synchronization code. Several reasons which contribute to false synchronization code are: a) The style of synchronization code, b) the length of synchronization code, b) the probability of “0” and “1” in synchronization code. Robustness of the synchronization also depend on the length of the synchronization code, longer is code more robust it is.

Barkers codes embedded in the proposed scheme as synchronization code satisfy above mentioned criteria to prevent false synchronization. A Barker code is a sequence of N values of +1 and −1, aj for j= 1, 2, 3 . . ., N.

such that:

for all

Here is a table of all known Barker codes, where negations and reversals of the codes have been omitted. A Barker code has a maximum [autocorrelation](http://en.wikipedia.org/wiki/Autocorrelation) of 1 (when codes are not aligned). Longer Barker-like codes exist; there is a 28 baud sequence which has sidelobes no larger than 2, and which thus has better RMS performance than the codes below. The table below shows all known Barker codes; it is conjectured that no other perfect binary phase codes exist.

|  |  |  |
| --- | --- | --- |
| **Known Barker Codes** | | |
| **Length** | **Codes** | |
| 2 | +1 −1 | +1 +1 |
| 3 | +1 +1 −1 | |
| 4 | +1 +1 −1 +1 | +1 +1 +1 −1 |
| 5 | +1 +1 +1 −1 +1 | |
| 7 | +1 +1 +1 −1 −1 +1 −1 | |
| 11 | +1 +1 +1 −1 −1 −1 +1 −1 −1 +1 −1 | |
| 13 | +1 +1 +1 +1 +1 −1 −1 +1 +1 −1 +1 −1 +1 | |

Barker codes of length 11 and 13 are used in [direct-sequence spread spectrum](http://en.wikipedia.org/wiki/Direct-sequence_spread_spectrum) and [pulse compression radar](http://en.wikipedia.org/wiki/Pulse_compression) systems because of their low [autocorrelation](http://en.wikipedia.org/wiki/Autocorrelation) properties.

The +ve and -ve amplitudes of the pulses forming the Barker codes imply the use of [biphase modulation](http://en.wikipedia.org/wiki/Biphase_modulation); that is, the [change of phase](http://en.wikipedia.org/wiki/Phase_shifting) in the [carrier wave](http://en.wikipedia.org/wiki/Carrier_wave) is 180 degrees.

A Barker code resembles a discrete version of a continuous [chirp](http://en.wikipedia.org/wiki/Chirp), another low-autocorrelation signal used in other pulse compression radars.

[Pseudorandom number sequences](http://en.wikipedia.org/wiki/Pseudorandom_number_sequence) can be thought of as cyclic Barker Codes, having perfect (and uniform) cyclic autocorrelation sidelobes. Very long pseudorandom number sequences can be constructed.

Similar to the Barker Codes are the [complementary sequences](http://en.wikipedia.org/wiki/Complementary_sequence), which cancel sidelobes exactly; the pair of 4 baud Barker Codes in the table form a complementary pair. There is a simple constructive method to create arbitrarily long complementary sequences.

In our algorithm we have taken barkers code length of 16 and all the “-1” are replaced by “0”.

4.4 Energy Efficient Watermark

As Jonathan el al[12] proposed that for a robust watermarking scheme, watermark should satisfy the Power Spectrum Condition, which state that power spectrum of watermark should be directly proportional to power spectrum of the original signal. In order to generate energy-efficient (PSC-compliant) watermarks, first, we evaluate the power spectrum of X of original audio signal x [1, ] by using periodogram,

Where is 2-D FFT of x [n1, n2].

Then we produce energy efficient watermark by

Where is 2-D FFT of the output of a unit variance white Gaussian random number generator.

4.5 QUANTIZATION INDEX MODULATION

Various information-embedding algorithms have been proposed. Cox Proposed a spread spectrum (SS) based system which embedded a pseudo-noise sequence concerning with watermark into the host signal. Malvar improved the SS algorithm [2]. This technique produced a dramatic improvement of the quality watermarking, because it created a watermark signal related to the cover image. Its performance of resisting additive noise is similar to that of the later-mentioned QIM method. However, both systems offer relatively little robustness when the host signal is unknown at the decoder. Wong proposed a novel QIM-like scheme which embedded information by modulating the component of a host vector in a given direction according to the embedded message. It has a good performance of attack resistance to JPEG compression, additive white Gaussian noise (AWGN) and low-pass filtering, yet it has a drawback of step size selection. QIM was first proposed by Chen and Womell [4]. It firstly embedded information by modulating an index or an index sequence with the watermark bits, and then quantized the host signal with an associated quantizer. Dithering was proposed at the same time to improve the performance and reduce the perceptual distortion in it. The conventional QIM used a fixed quantization step size. To promote the transparence and robustness, we alter the fixed quantization step size in QIM by an adaptive step size which is determined by perceptual contrast masking. Compared with the QIM scheme proposed by Li[5], our method applies a different strategy in embedded location and step-size decision. Experimental results show that our method is more robust to AWGN, salt & pepper noise, JPEG compression than the QIM.

The traditional QIM scheme firstly embeds a watermark bit by dither modulating a host index or an index sequence, and then quantizing the host signal with associated quantizers. The simplest case of QIM is embedding one bit in a single host sample. Let m ϵ {0, 1 } be a 1-bit watermark message, s ϵ R be a host sample, scalar uniform quantizer Q with step size □ is rounding a value. Hence the watermarked signal is generated by

(1)

where . or is an offset used for embedding bit "0" or "1" respectively. These two quantizers form two lattices

, (2)

Where, Z is the set of an integer.

At the detection stage, the corrupted signal y is received as y = x + v, where x is the watermarked signal, v denotes the noise resulted from intentional or unintentional attacks. The watermark extractor in QIM system is a minimum-distance detector. It calculates the nearest quantized point to y and outputs the estimated message bit:

mϵ {0, 1}

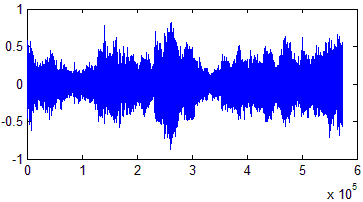
Where .

For the traditional QIM, the quantization step size is fixed without taking into account the image content which is known to embedder. To improve the transparence and robustness, we suggest a method based on the human visual system (HVS). HVS is less sensitive to changes in heavy texture regions than the changes occurred in smooth regions. Therefore, the quantization step size at each host vector could be automatically selected by using a perceptual model.

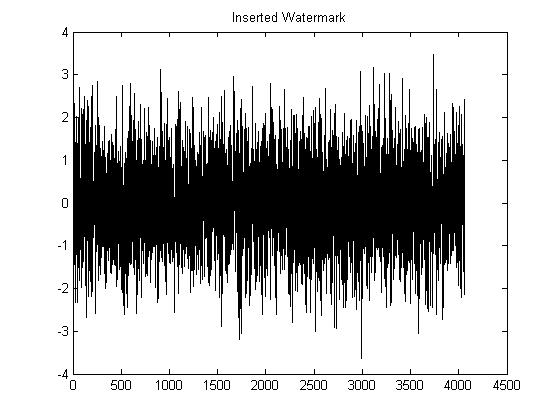
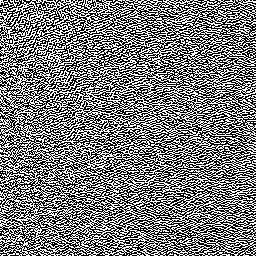
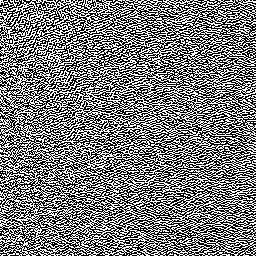
4.6 SPREAD SPECTRUM WATERMARKING PROCEDURE (Cox algorithm)

4.6.1 Inserting the Watermark

Fig.4.2 illustrates the general procedure for frequency domain watermarking. Upon applying a frequency transformation to the data, a perceptual mask is computed that highlights perceptually significant regions in the spectrum that can support the watermark without affecting perceptual fidelity. The watermark signal is then inserted into these regions in a manner described in Section 4.4. The precise magnitude of each modification is only known to the owner.



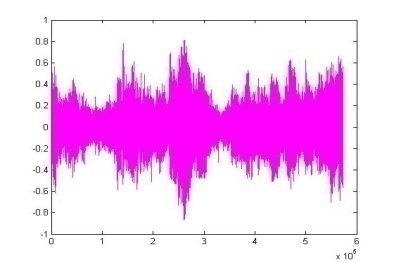
**Cover Audio**



**FFT/DCT**

**Determine Perceptually Significant Regions**

**Inverse FFT/DCT**



**Watermarked Audio**

**Insert Watermark**

**Figure 4.1 Stages of watermark insertion process.**

By contrast, an attacker may only have knowledge of the possible range of modification. To be confident of eliminating a watermark, an attacker must assume that each modification was at the limit of this range, despite the fact that few such modifications are typically this large. As a result, an attack creates visible (or audible) defects in the data. Similarly, unintentional signal distortions due to compression or image manipulation must leave the perceptually significant spectral components intact; otherwise the resulting image will be severely degraded. This is why the watermark is robust.

When we insert X into V to obtain V’ we specify a scaling parameter α, which determines the extent to which X alters V. Three natural formulae for computing V’ are

(4.1)

(4.2)

(4.3)

Equation (4.1) is always invertible, and (4.2) and (4.3) are invertible if Vi != 0, which holds in all of our experiments. Given V\*, we can therefore compute the inverse function to derive X\* from V\* and V.

Equation (4.1) may not be appropriate when the values vary widely. If Vi = 106, then adding 100 may be insufficient for establishing a mark, but if Vi =10 adding 100 will distort this value unacceptably. Insertion based on (4.2) or (4.2) are more robust against such differences in scale. We note that (4.2) and (4.2) give similar results when αxi is small. Also, when Vi is positive, then (4.2) is equivalent to lg(Vi’) = lg(Vi) + αxi, and may be viewed as an application of (4.1) to the case where the logarithms of the original values are used.

4.6.2 Determining Scaling Parameter (α)

A single scaling parameter α may not be applicable for perturbing all of the values Vi, since different spectral components may exhibit more or less tolerance to modification. More generally one can have multiple scaling parameters α1 ,…., αn and use update rules such as . We can view αi as a relative measure of how much one must alter Vi to alter the perceptual quality of the document. A large αi means that one can perceptually “get-away” with altering Vi, by a large factor without degrading the document.

4.6.3 Choosing the Length (n), of the Watermark

The choice of n dictates the degree to which the watermark is spread out among the relevant components of the image. In general, as the number of altered components is increased the extent to which they must be altered decreases. For a more quantitative assessment of this tradeoff, we consider watermarks of the form and model a white noise attack by Vi\* = Vi’ + ri where ri are chosen according to independent normal distributions with standard deviation σ. For the watermarking procedure described below, one can recover the watermark when α is proportional to.

Note that the number of bits of information associated with the watermark can be arbitrary. The watermark is simply used as an index to a database entry associated with the watermark.

4.6.4 Extracting the Watermark

The procedure for extraction and decoding of the watermark is shown in figure 4.3. We extract from each image or document D a sequence of values V = V1, …, Vn; into which we insert a watermark X = x1, …, xn, to obtain an adjusted sequence of values V’ = v1’ …… vn’. V’ is then inserted back into the document in place of V to obtain a watermarked document D’. One or more attackers may then alter D’, producing a new document D\*. Given D and D\*, a possibly corrupted watermark X\* is extracted and is compared to X for statistical significance. We extract X\* by first extracting a set of values V\* = v1\* …… vn\* from D\* (using information about D) and then generating X\* from V\* and V. Frequency-domain based methods for extracting V\* and V and inserting are given in Section 4.4.

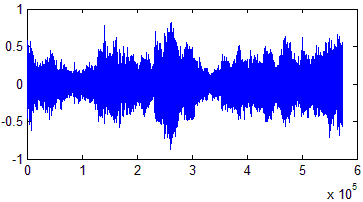
4.6.5 Evaluating the Similarity of Watermarks

It is highly unlikely that the extracted mark X\* will be identical to the original watermark X. Even the act of re-quantizing the watermarked document for delivery will cause X\* to deviate from X. We measure the similarity of X\* and X by



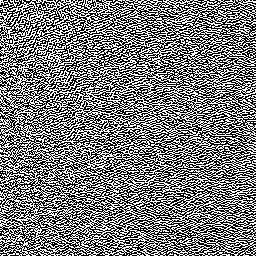
(4.4)

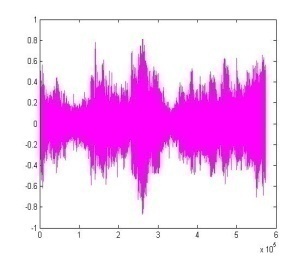
Many other measures are possible, including the standard correlation coefficient. To decide whether X and X\* match, one determines whether>T, where T is some threshold. Setting the detection threshold is a classical decision estimation problem in which we wish to minimize both the rate of false negatives (missed detections) and false positives (false alarms). I.J. Cox et al. [1] have chosen this measure so that it is particularly easy to determine the probability of false positives.



**FFT/DCT**

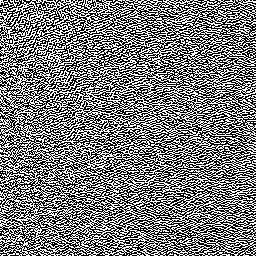
**Cover Audio**



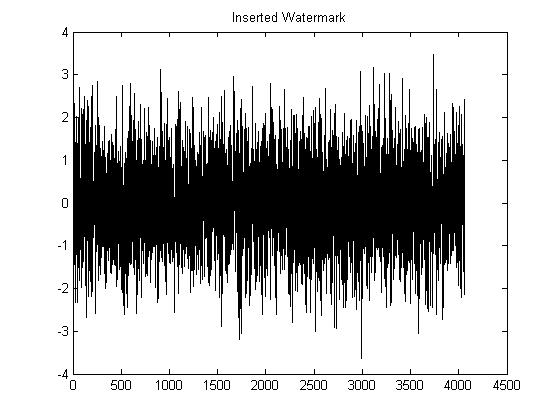
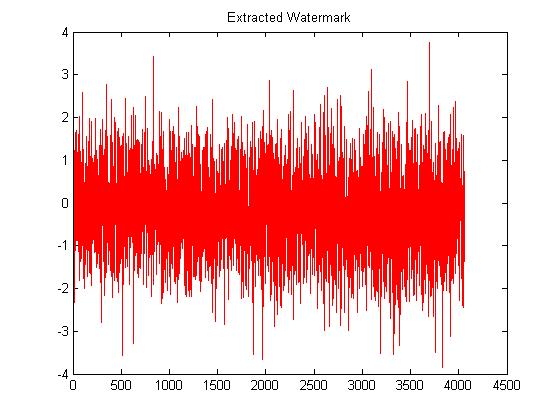


**Watermarked Audio**

**FFT/DCT**



**Extracted Watermark(X\*)**



**Similar?**

**Sim(X, X\*)**

**Original Watermark(X)**

**Figure 4.2 Extraction and decoding of the Watermark.**

4.7 Proposed Algorithm for Watermark Embedding

In order to guarantee robustness and transparency of watermarking, the proposed scheme embeds synchronization code in the mean value of several samples Let Y = {y (i); 0 ≤ i < Length} represent a host digital audio signal with Length samples. W = { w( i , j ); 0 ≤ i < M; 0 ≤ j <N } is a binary image to be embedded in lower frequency of the host audio signal, and w (i, j) ϵ (0, 1) is the pixel value at (i, j). W1 is the energy efficient watermark which is generated in each segment by second lowest frequency component and embedded in the same. F ={ f( i ); 0 ≤ i < Lsyn } is a synchronization code with Lsyn bits, where f(i) ϵ (0, 1). The main steps of the embedding procedure developed can be described as follows.

Let Y0 represent each segment and is cut into two sections and with L1 and L2 samples respectively. Synchronization code and watermarks are embedded into and , respectively.

4.7.1 Synchronization Code Embedding

The synchronization code embedding proceeds as follows:

1. Audio segment is cut into L*sync* audio segments, and each audio segment having n sample, where

1. Calculate the mean value of , that is
2. The synchronization code can be embedded into each by quantizing the mean value , the rule is given by

Where is the original signal sample and is modified sample and

Where mod (x , y) returns the remainder of division of *x* by *y* , and S1 is the quantization step.

4.7.2 Watermark Embedding

1. For each audio segment , H level DWT is performed to get the wavelet coefficient of , ,, . . . , , where is coarse signal and the detail signal are , . . . , , .
2. To take the advantage of lower frequency coefficient which has a higher energy value and robust against various signal processing, the DCT is performed on only lower frequency coefficient
3. *Watermark Embedding at Lower frequency component*: For robustness and transparency of watermark at first level the watermark is embedded in magnitude of DCT coefficient by quantization. The proposed algorithm embeds all watermark bit in each segment of the audio signal. The quantization function for embedding is as follows:

t = (i-1) × N + j

Where 0 ≤ i < M, 0 ≤ j < N and is quantization step, and

1. *Watermark Embedding at second lowest frequency component:* In second level of watermarking second lowest frequency component is used for watermarking. is used to generate energy efficient watermark and then the generated watermark is embedded in using cox algorithm.

4.7.3 Generation of Energy Efficient Watermark

1. Convert into two dimensional square matrix *x*[ n1, n2 ]. Where n1=n2= size of square matrix.
2. Find

Where is 2-D FFT of *x*[ n1, n2 ].

1. Find

which is the 2-D FFT of .

Where is the output of unit-variance white Gaussian random number generator.

1. Find . such that
2. Convert to one dimension where m =
3. . Energy Efficient watermark W =

Where SD is Standard Deviation of

4.7.4 Embedding of Energy Efficient Watermark

To embedding the above generated watermark Cox algorithm is used. Which state that when we insert X into V to obtain we specify a scaling parameter α, which determines the extent to which X alters V. Three natural formulae for computing are

(1)

(2)

(3)

Equation (1) is always invertible, and (2) and (3) are invertible if , which holds in all of our experiments. Given, we can therefore compute the inverse function to derive from and *V*.

The watermark is embedded in two dimensional *x*[ n1, n2 ] and then watermarked

*x’*[ n1, n2 ] is converted back to one dimensional .

1. To obtain the lower frequency coefficient of the signal, Inverse DCT is performed on lower frequency coefficient
2. H level inverse DWT is performed after replacing with and with and then the watermarked audio signal is

4.8 Repeat Embedding

In order to improve robustness against cropping Section III-A and III-B is repeated to embed synchronization code and watermarks into every sections.

In our experiment the length of watermark embedding is fixed as:

4.9 Watermarks detection

4.9.1 First level watermark detection

The watermark detection at first level in proposed scheme neither needs the original audio nor any side information. The water marking detection procedure is stated as follows:

1. Watermark segment position *L* is located based on the frame synchronization technology of digital communications.
2. H- level DWT is performed on each audio segment *Y\**(*m*) after *L* and then get the coefficient as follows:
3. DCT is performed on lower frequency DWT- coefficient

1. Rule for extraction is

4.9.2 Second level watermark detection

The watermark detection at second level requires original audio since the Cox algorithm used to embed the watermark is a blind technique. If we use equation (2) as mentioned above than the inverse function corresponding to it is:

The inverse of the other two functions are:

Any of the three functions can be used to detect the watermark according to the choice of embedding function used.

4.10 Similarity Functions

**For First level:**

The Similarity Function for first level is :

Where is the original watermark and is the extracted watermark [33].

It is highly unlikely that the extracted mark will be identical to the original mark. Even the act of requantization the watermark document for delivery will cause to deviate from

**For second level:**

As discussed above in Cox algorithm the similarity function used to detect the correlation between extracted and inserted energy efficient random watermark is given by:

To decide whether and match, one determines whether sim( .) >T, where T is some threshold. Setting the detection threshold is a classical decision estimation problem in which we wish to minimize both the rate of false negatives (missed detections) and false positives (false alarms).

4.11 Signal to Noise Ratio

For audio signal quality signal to noise ratio (SNR) is defined as follows:

**Chapter 5**

**RESULTS AND DISCUSSION**

In order to illustrate the inaudibility and robust nature of proposed watermarking scheme, the watermarks are applied to two audio piece. Audio signal in test are 16 bit signed mono audio signal sampled at 44.1 KHz. 64×64 bit binary image is used as first level watermark for all audio signal and 16 bit barker code 1111100110101110 as synchronization code. The Doubechies-1 wavelet basis is used. The smaller level influence robustness of the watermark and larger level will cause large calculation, so 4-levelDWT is performed in this test. In our experiment synchronization code is embedded in mean value of five samples and 65536 samples audio segment is used for watermark embedding.

The experiment is performed on two audio files samples shown in figure 5.1(a) and 5.1(b). The lengths of samples are 13.03s and 19.01s, respectively. The quantization step and alpha for two samples are S1 = S2 = alpha=0 .035 and S1 = S2 = alpha= 0.05 respectively.

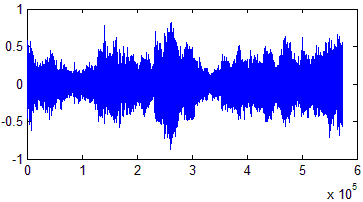
Samples

Figure 5.1 (a) Test Audio sample 1

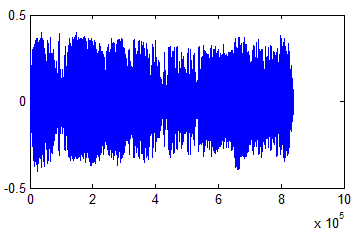
Samples

Figure 5.1 (b) Test Audio samples 2

Figure 5.2 Show the binary 64×64 binary image of three which is embedded in the first level of the audio using QIM.



Figure 5.2 Binary image used as watermark for first level.

Figure 5.3 shows one of the random generated energy efficient watermark during the experiment for the test audio sample 1.

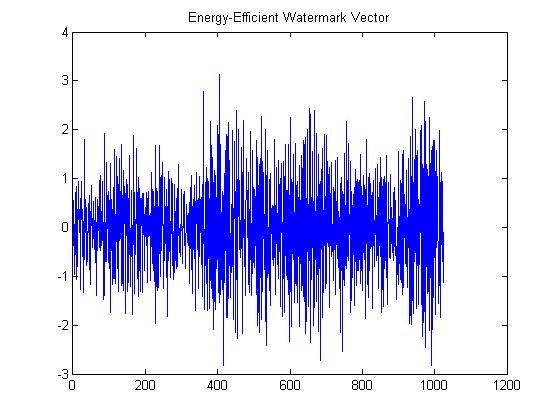


Figure 5.3 A random energy efficient watermark generated during the experiment.

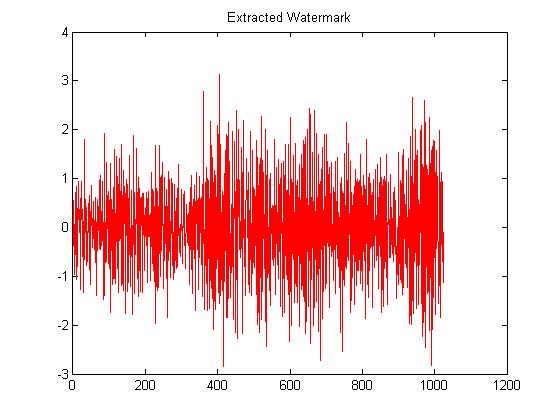
Figure 5.4 Shows the extracted watermark of the above embedded watermark during the extraction process as we can see that both the watermarks are similar and the similarity factor for the same will be discussed further in the chapter. 

Figure 5.4 Extracted random watermark.

The experimental result for the test audio sample 1 is shown in figure form the below figures for various attacks.

5.1 ORIGINAL WATERMARKED AUDIO WITHOUT ANY ATTACK

Figure 5.5 show the original audio file, figure 5.6 show the watermarked file, figure 5.7 show the difference between the watermarked and original file so we can observe the difference between two. Figure 5.8(a) and figure 5.8(b) show the detector response for the first level and second level.

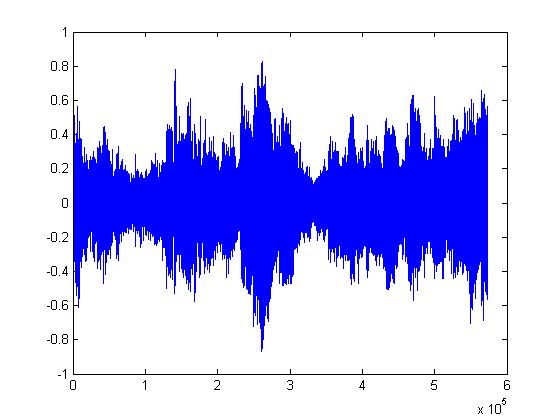


Figure 5.5 Original Audio File

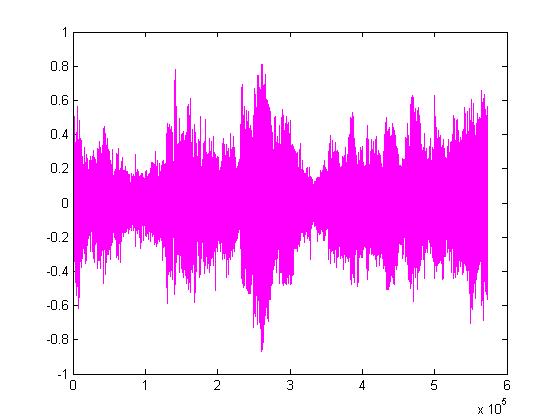


Figure 5.6 Watermarked Audio File

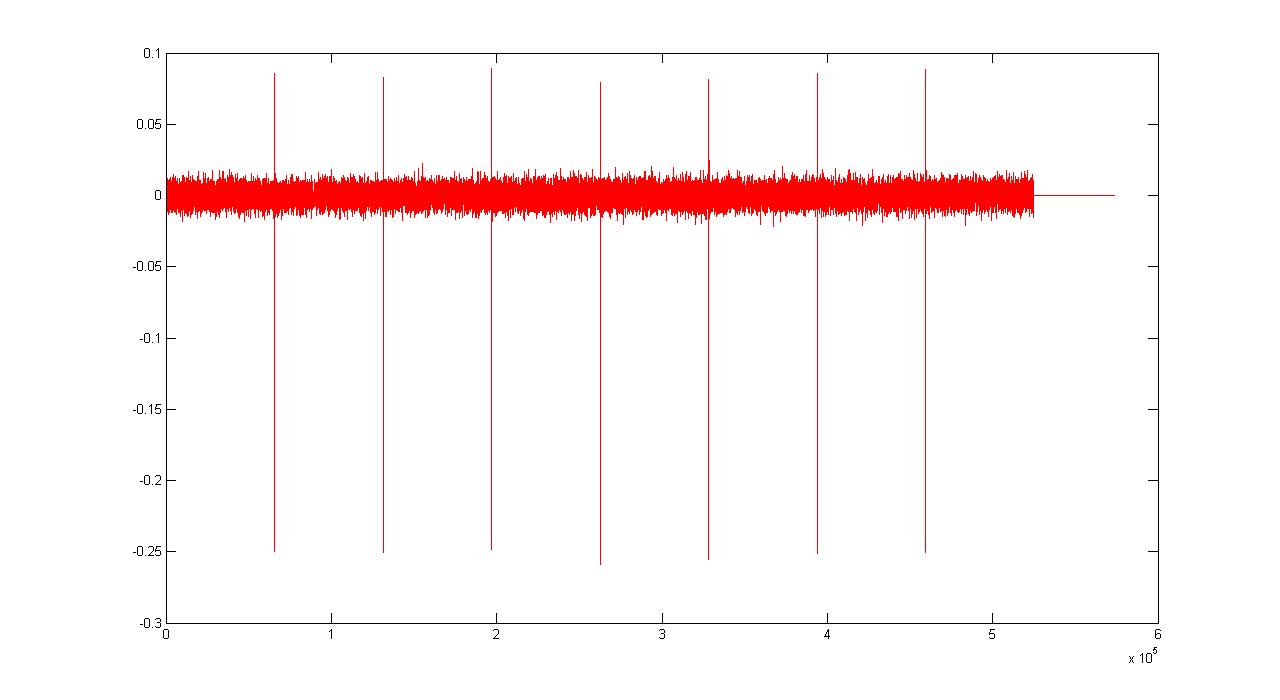


Figure 5.7 Difference between original and watermark

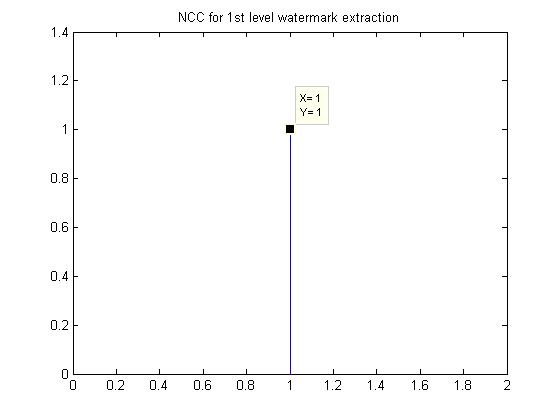
 

Figure 5.8 (a) Extracted watermark and NCC of the watermark at 1st level in original Audio

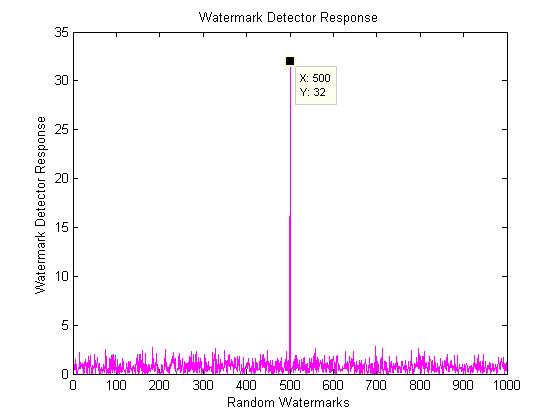


Figure 5.8 (b) Similarity of 2nd level watermarks in Original Audio

5.2 Robustness test

To illustrate the robustness of our watermarking scheme, the attack including MPEG compression, resampling, requantization and noise adding are used. The various attacks can be described as:

5.2.1 Resampling

In this experiment the original signals are sampled with a sampling rate of 44.1 KHz. Watermarked audio signals are down sampled to 22.05 KHz, 11.025 KHz, 8.82 KHz and back to 44.1 KHz.

**Experiment result of Resampling watermarked audio file to 22.05 KHz.**

Figure 5.9(a) show the watermarked file after resampling to 22.05 KHz and figure 5.9(b), figure 5.9(c) show the detector response at first level and second level respectively.

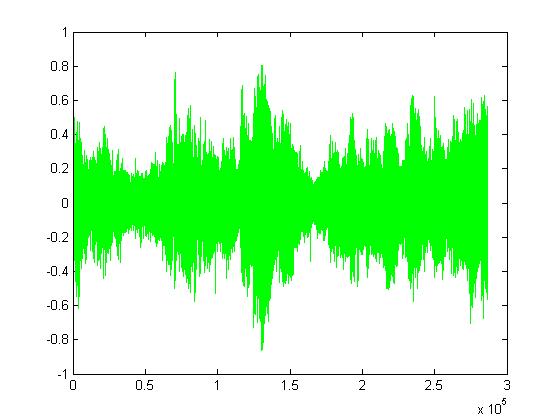


Figure 5.9 (a) Audio file Resample to 22.05 KHz

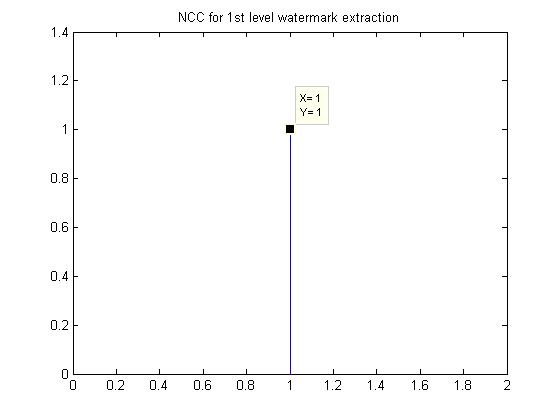
 

Figure 5.9 (b) Extracted watermark and NCC of the watermark at 1st level in resampling to 22.05 KHz

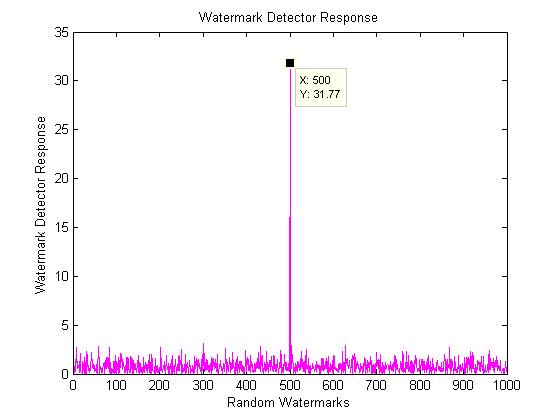


Figure 5.9 (c) Similarity of 2nd level watermarks in sampling to 22.05 KHz.

**Experiment result of Resampling watermarked audio file to 11.025 KHz**

Figure 5.10(a) show the watermarked file after resampling to 11.025 KHz and figure 5.10(b), figure 5.10(c) show the detector response at first level and second level respectively.

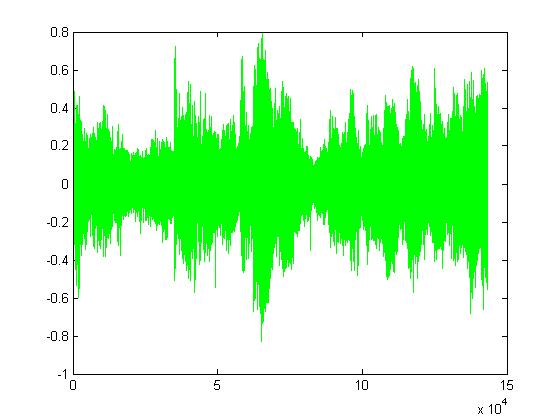


Figure 5.10 (a) Audio file Resample to 11.025 KHz

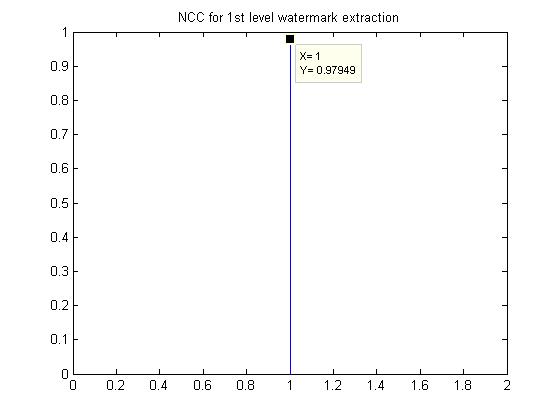
 

Figure 5.10 (b) Extracted watermark and NCC of the watermark at 1st level in resampling to 11.025 KHz

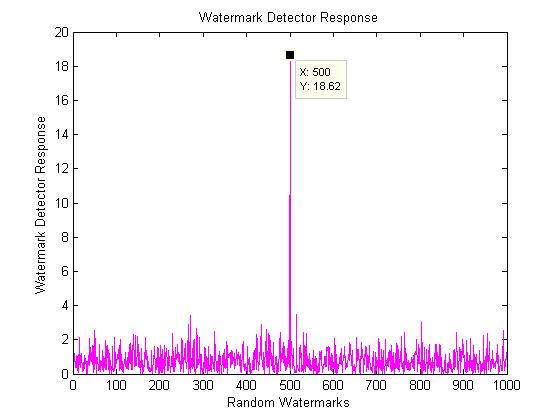


Figure 5.10 (c) Similarity of 2nd level watermarks in sampling to 11.025 KHz**.**

**Experiment result of Resampling watermarked audio file to 8.82 KHz**

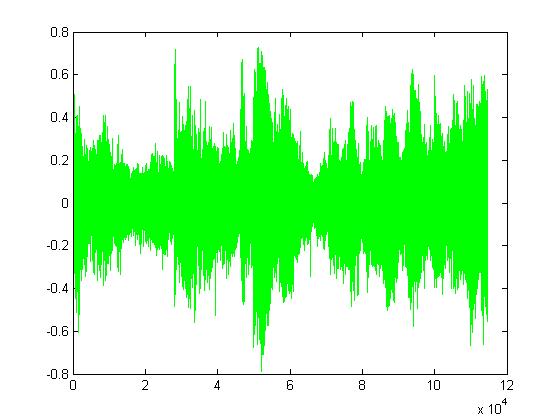
Figure 5.10(a) show the watermarked file after resampling to 8.825 KHz and figure 5.11(b), figure 5.11(c) show the detector response at first level and second level respectively. 

Figure 5.11 (a) Audio file Resample to 8.82 KHz

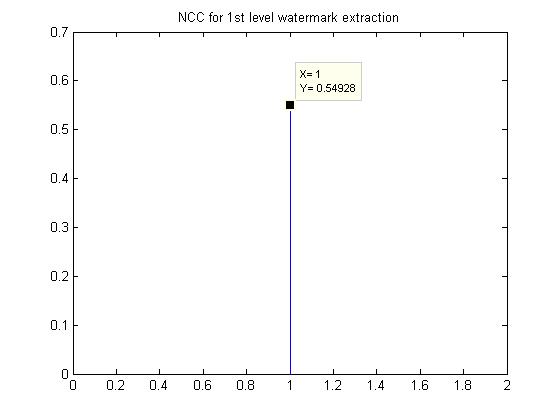
 

Figure 5.11 (b) Extracted watermark and NCC of the watermark at 1st level in resampling to 8.82 KHz

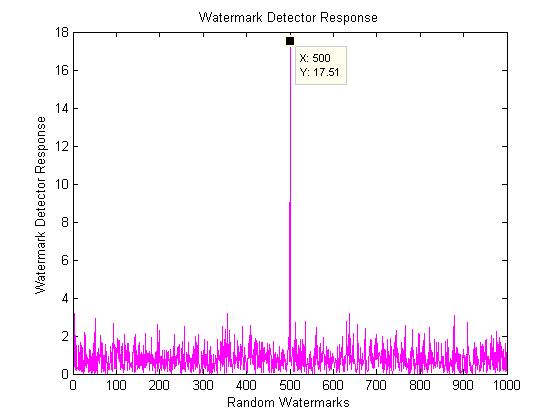


Figure5.11 (c) Similarity of 2nd level watermarks in sampling to 8.82 KHz.

5.2.2 Requantization

Audio tracks sampled at 8-bit are often used in the game and multimedia application. We therefore tested the process of requantization of 16-bit watermarked audio signal to 8-bit and back to 16-bit.

**Experiment result of Requantization of watermarked audio file to 8-bit**

Figure 5.12(a) show the watermarked file after requantization to 8 bit and figure 5.12(b), figure 5.12(c) show the detector response at first level and second level respectively.

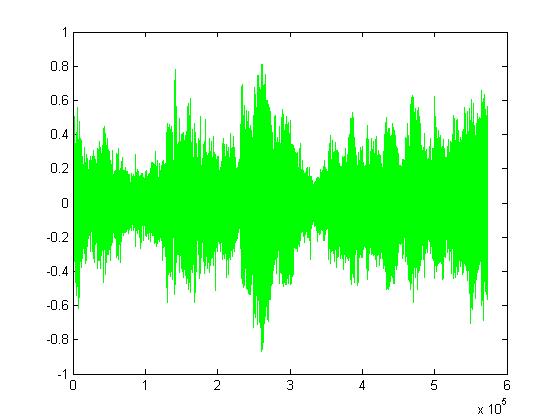


Figure 5.12 (a) Audio file after Requantization to 8 bit

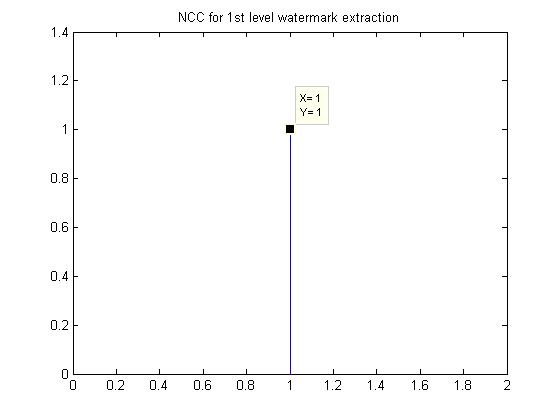
 

Figure 5.12 (b) Extracted watermark and NCC of the watermark at 1st level in Requantization to 8 bit

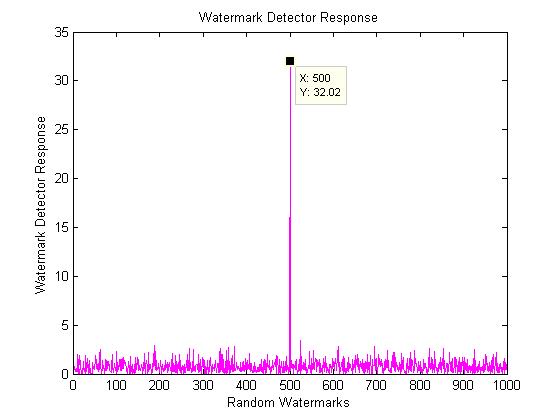


Figure 5.12 (c) Similarity of 2nd level watermarks in Requantization to 8 bit

5.2.3 Additive noise

White noise with 10% of the power of the audio signal is added.

Figure 5.13(a) show the watermarked file after addition of noise and figure 5.13(b), figure 5.9(c) show the detector response at first level and second level respectively.

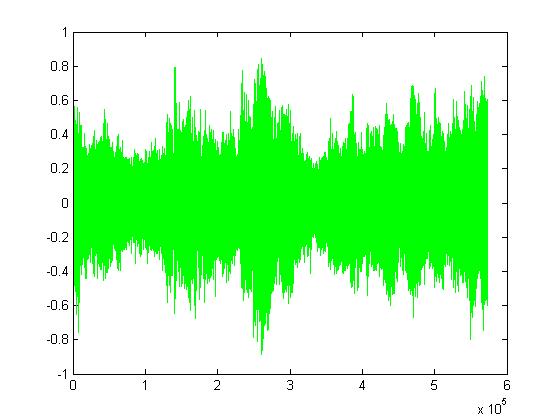


Figure 5.13 (a) Audio file after addition of noise

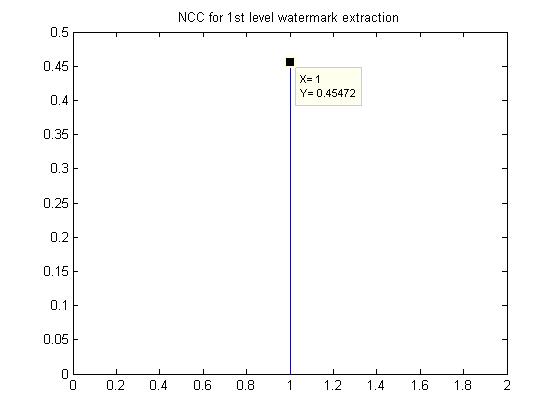
 

Figure 5.13 (b) Extracted watermark and NCC of the watermark at 1st level after additive noise

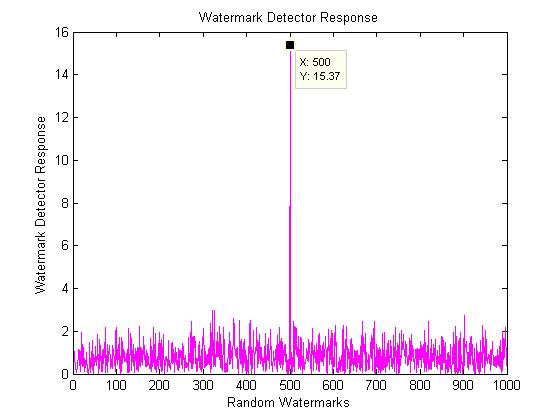


Figure 5.13 (c) Similarity of 2nd level watermarks after additive noise

5.2.4 MPEG compression

We have studied the robustness of the watermark to coding/decoding of the audio signal. The coding/decoding is performed using a software implementation of the ISO/MPEG-2 Audio Layer III coder with several different bit rates (128, 112, 64, 48, 32 kb).

**Experiment result for Mp3 compression at 128 Kb.**

Figure 5.14(a) show the watermarked file after Mp3 compression at 128 Kb and figure 5.14(b), figure 5.14(c) show the detector response at first level and second level respectively.

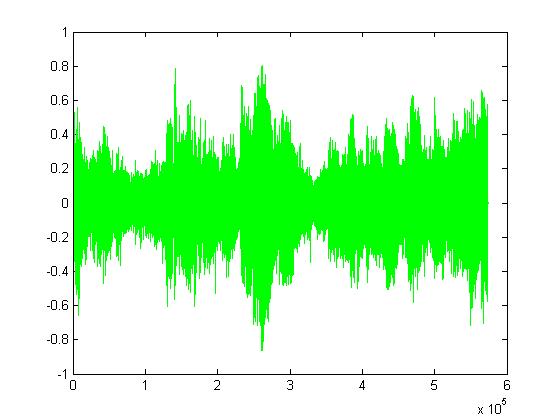


Figure 5.14 (a) Audio file after Mp3 compression at 128 Kb

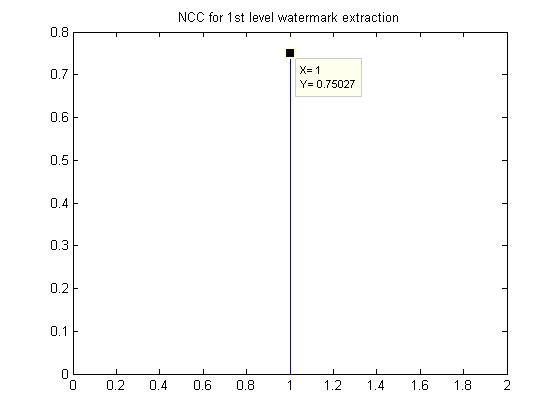
 

Figure 5.14 (b) Extracted watermark and NCC of the watermark at 1st level after Mp3 at 128 kb

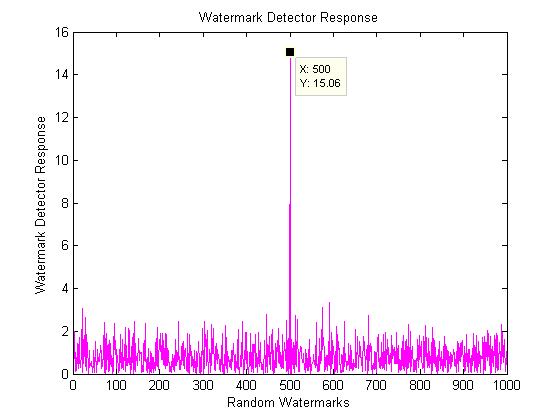


Figure 5.14 (c) Similarity of 2nd level watermarks after Mp3 at 128 kb

**Experiment result for Mp3 compression at 112 Kb.**

Figure 5.15(a) show the watermarked file after Mp3 compression at 112 Kb and figure 5.15(b), figure 5.15(c) show the detector response at first level and second level respectively.

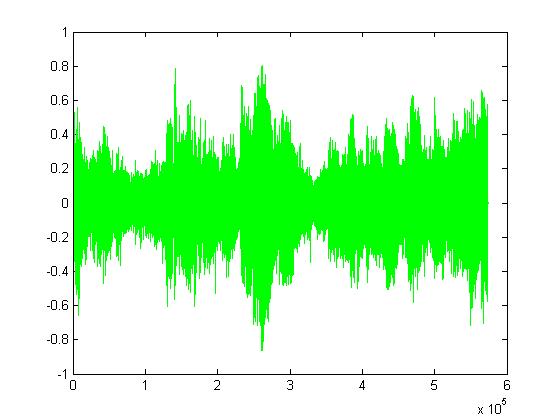


Figure 5.15 (a) Audio file after Mp3 compression at 112 Kb

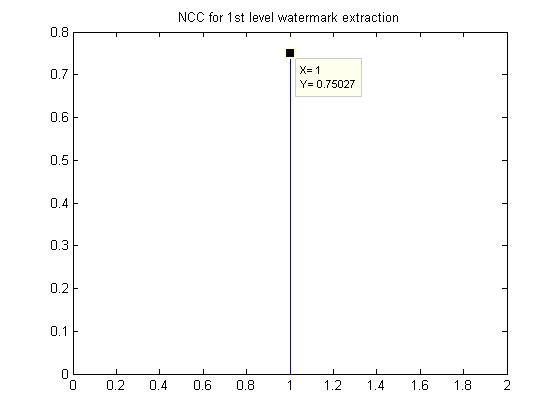
 

Figure 5.15 (b) Extracted watermark and NCC of the watermark at 1st level after Mp3 at 112 Kb

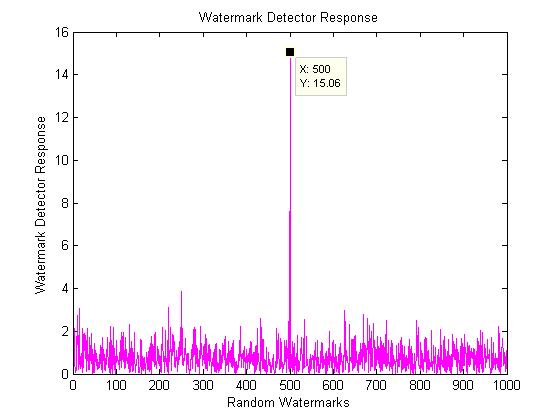


Figure 5.15 (c) Similarity of 2nd level watermarks after Mp3 at 112 Kb

**Experiment result for Mp3 compression at 64 Kb.**

Figure 5.16(a) show the watermarked file after Mp3 compression at 64 Kb and figure 5.16(b), figure 5.16(c) show the detector response at first level and second level respectively.

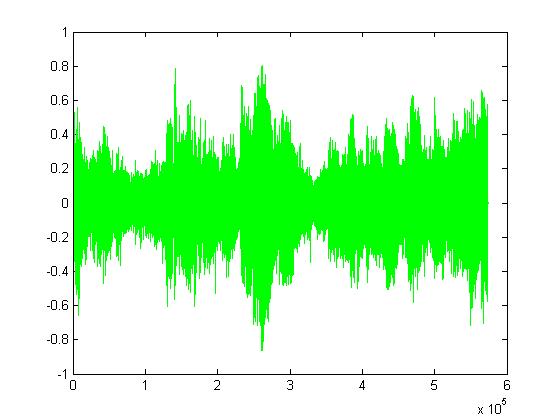


Figure 5.16 (a) Audio file after Mp3 compression at 64 Kb

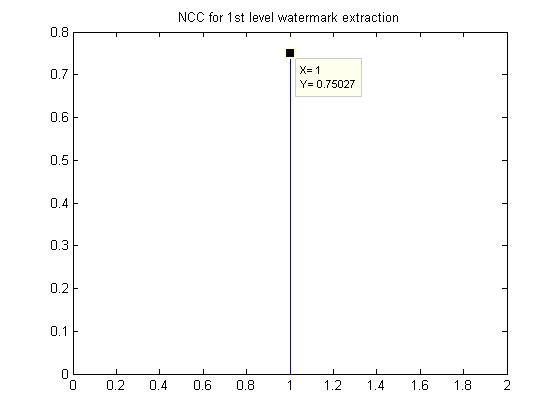
 

Figure 5.16 (b) Extracted watermark and NCC of the watermark at 1st level after Mp3 at 64 Kb

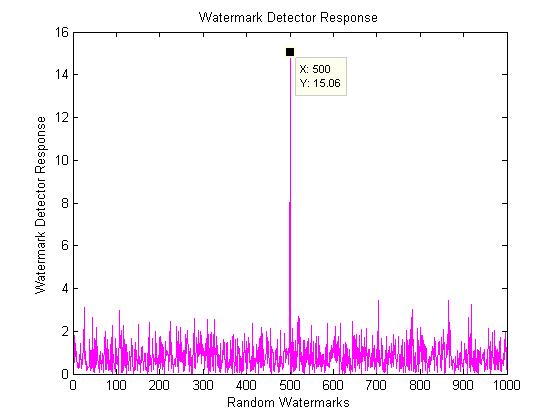


Figure 5.16 (c) Similarity of 2nd level watermarks after Mp3 at 64 Kb

**Experiment result for Mp3 compression at 48 Kb**

Figure 5.17(a) show the watermarked file after Mp3 compression at 48 Kb and figure 5.17(b), figure 5.17(c) show the detector response at first level and second level respectively.

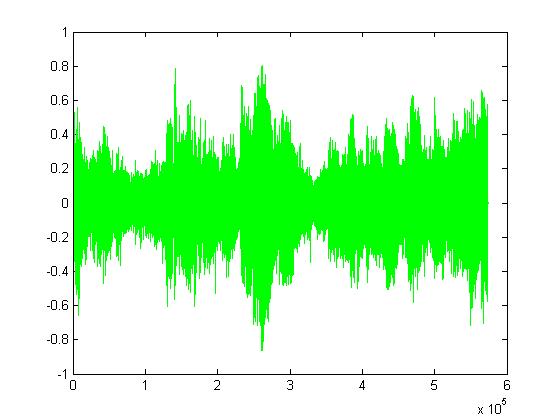


Figure 5.17 (a) Audio file after Mp3 compression at 48 Kb

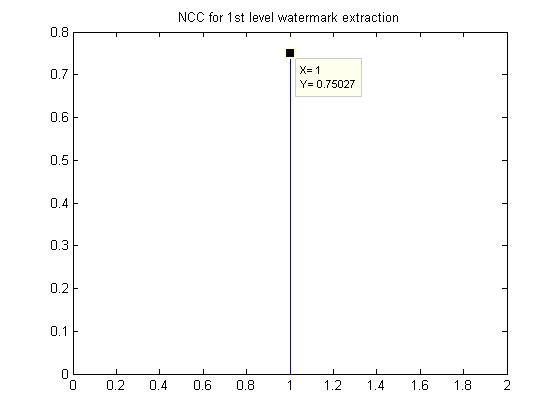
 

Figure 5.17 (b) Extracted watermark and NCC of the watermark at 1st level after Mp3 at 48 Kb

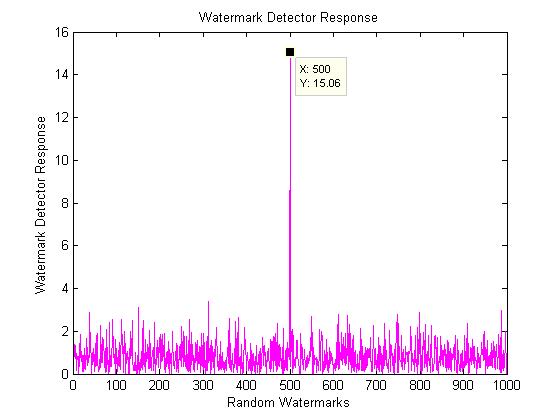


Figure 5.17 (c) Similarity of 2nd level watermarks after Mp3 at 48 Kb

**Experiment result for Mp3 compression at 32 Kb.**

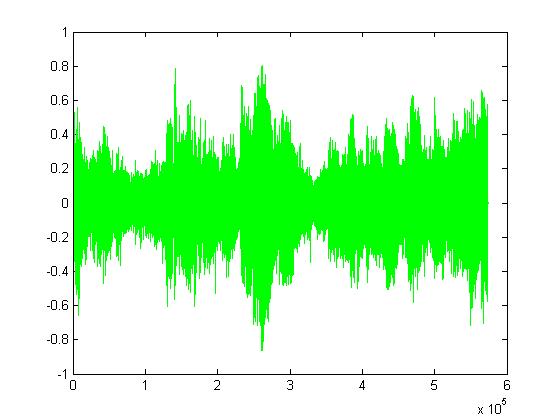
Figure 5.18(a) show the watermarked file after Mp3 compression at 32 Kb and figure 5.18(b), figure 5.18(c) show the detector response at first level and second level respectively.

Figure 5.18 (a) Audio file after Mp3 compression at 32 Kb.

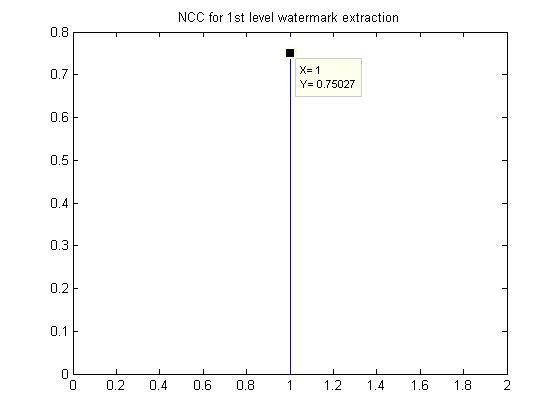
 

Figure 5.16 (b) Extracted watermark and NCC of the watermark at 1st level after Mp3 at 32 Kb.

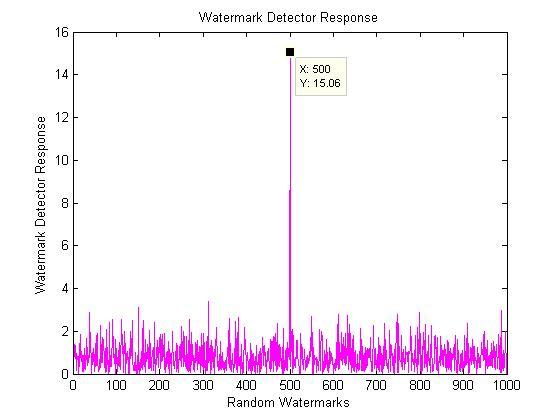


Figure 5.18 (c) Similarity of 2nd level watermarks after Mp3 at 32 Kb.

Table 5.1 Detectors response for various attacks on sample 1 and sample 2 at first and second level

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attack type | Test sample 1 | | Test sample 2 | |
| 1st level of SIM 1 | 2nd level of  SIM 2 | 1st level of SIM 1 | 2nd level of SIM 2 |
| Original watermark | 1.0000 | 32.0089 | 1.0000 | 32.0147 |
| Mp3-128k | 0.7503 | 15.0641 | 0.8935 | 22.4879 |
| Mp3-112k | 0.7503 | 15.0641 | 0.8935 | 22.4879 |
| Mp3-64k | 0.7503 | 15.0641 | 0.8935 | 22.4879 |
| Mp3-48k | 0.7503 | 15.0641 | 0.8935 | 22.4879 |
| Mp3-32k | 0.7503 | 15.0641 | 0.8935 | 22.4879 |
| Additive noise | 0.4547 | 15.3689 | 0.5287 | 14.288 |
| Re-quantization | 1.0000 | 32.0221 | 1.0000 | 32.0147 |
| Re-sampling 22.05 KHz | 1.0000 | 31.7693 | 1.0000 | 31.968 |
| Re-sampling 11.025 KHz | 0.9795 | 18.6173 | 1.0000 | 30.6748 |
| Re-sampling 8 KHz | 0.5493 | 17.5146 | 1.0000 | 29.2121 |
| Cropping | 1.0000 | 5.1765 | 1.0000 | 5.2349 |

Table 5.1 show the detectors response for audio sample 1 and audio sample 2 for various attacks. The quantization step and alpha for two samples are S1 = S2 = alpha=0 .035 and S1 = S2 = alpha= 0.05 respectively.

**Chapter 6**

**CONCLUSIONS AND FUTURE DIRECTIONS**

On the basis of energy efficient watermark and synchronization we proposed a robust audio watermarking scheme. To increase the robustness of synchronization code the proposed algorithm Barkers code as synchronization code, embedded in mean value of the several samples of audio and embedding one watermark in DWT-DCT coefficient and generating other energy efficient watermark which is embedded in DWT coefficient. Dual water marking scheme used here make the scheme highly robust as both watermark compensate the drawback of each other.

The key result is the power-spectrum condition (PSC), which states that a watermark is energy-efficient if and only if its power spectrum is directly proportional to that of the original signal. The watermark must be designed in accordance to the power spectrum of the original cover image. The PSC holds for any signals that meet the assumptions of the model. It may therefore be applicable to digital audio, images, and video for example.

Synchronization is one of the key issues of audio watermarking and losing synchronization cause false detection. Time scale and frequency scale modification can cause lost of synchronization. So there is need of an exact synchronization algorithm based on robust synchronization code. In our algorithm the barkers code used prove to be highly robust to the synchronization attack.

To illustrate the robustness of our watermarking scheme, the attack including MPEG compression, resampling, requantization and noise adding are used.

The experimental results have illustrated the robustness and inaudible nature of the audio watermark.

FUTURE WORK

I have implemented a dual watermarking scheme to embed the watermark. Although results has not been that anticipated but if reviewed again giving full attention on where the two procedures could conflict, it will result in a very robust watermarking technique resilient to almost all categories of attacks.

Despite the success of the proposed algorithm for watermarking, it also has draw back. In this method the second level water mark cannot be extracted in cropping attack, so PSC compliant method cannot be used alone to resist cropping attack. Due to dual watermarking technique which may become complex for practical implementation of encoder and decoder also likely to pose problem for real time application, Further research will focus on overcoming this problem.

**Chapter 7**

**BIBLIOGRAPHY**

1. I. J. Cox and M. L. Miller, “*The first 50 years of electronic watermarking*,” J. Appl. Signal Process., vol. 56, no. 2, pp. 225–230, 2002.
2. L. Wei, Y. Yi-Qun, L. Xiao-Qiang, X. Xiang-Yang, and L. Pei-Zhong, “*Overview of digital audio watermarking*,” J. Commun., vol. 26, no. 2, pp. 100–111, 2005.
3. L. Wen-Nung and C. Li-Chun, “*Robust and high-quality time-domain audio watermarking subject to psycho acoustic masking,*” in Proc. IEEE Int. Symp. Circuits Syst., AZ, 2002, vol. 2, pp. 45–48.
4. D. Megías, J. Herrera-joancomartí, and J. Minguillón, “*A robust audio watermarking scheme based on MPEG 1 layer III compression*,” in Communications and Multimedia Security—CMS 2003. New York: Springer-Verlag, 2003, vol. 963, LNCS, pp. 226–238.
5. H. J. Kim, “*Audio watermarking techniques*,” in Pacific Rim Workshop on Digital Steganography, Kyushu Institute of Technology, Kitakyushu, Japan, Jul. 3–4, 2003.
6. S. Sheng-He, L. Zhe-Ming, and N. Xia-Mu, Digital Watermarking Technique. Beijing, China: Science, 2004.
7. D. Kirovski and H. S. Malvar, “Spread spectrumwatermarking of audio signals,” IEEE Trans. Signal Process., vol. 51, no. 4, pp. 1020–1033, Apr. 2003.
8. H. O. Kim, B. K. Lee, and N. Y. Lee, Wavelet-based audio watermarking techniques: Robustness and fast synchronization [Online]. Available: *http://amath.kaist.ac.kr/research/paper/01–11.pdf*
9. W. Yong, H. Ji-Wu, and Y. Q. Shi, “*Meaningful watermarking for audio with fast resynchronization,*” J. Comput. Res. Develop., vol. 40, no. 20, pp. 215–220, 2003.
10. J. W. Huang, W. Yong, and Y. Q. Shi, “*A blind audio watermarking algorithm with self-synchronization*,” in Proc. IEEE Int. Symp. Circuits Syst., AZ, 2002, vol. 3, pp. 627–630.
11. X.Y.Wang and H.Zhao*, “* [*A Novel Synchronization Invariant Audio Watermarking Scheme Based on DWT and DCT”*](http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=4014398&queryText%3DSynchronization+Invariant+Audio+Watermarking+SchemeT+and+DCT%26openedRefinements%3D*%26searchField%3DSearch+All) IEEE Transactions on Signal Processing, vol. 54, no.12, pp.4835-4840, 2006
12. Su, J.K.; Girod, B**., *“****Power-spectrum condition for energy-efficient watermarking”* IEEETransactions on Multimedia,vol. 4, no. 4, pp. 551-560, 2002
13. I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, “*Secure spread spectrum watermarking for multimedia*,” NEC Res. Inst., Princeton, NJ, Tech. Rep. 95-10, Oct. 1995.
14. Bassia, P.; Pitas, I. 1998. Robust Audio Watermarking in the time-domain, to appear in Proc. of EUSIPCO'98, Rhodes, Greece.
15. Neubauer, C.; Herre, J. 1998. Digital Watermarking and its Influence on Audio Quality. 105th AES Convention, San Fransisco, California. Preprint 4823.
16. Neubauer, C.; Herre, J.; Brandenburg, K. 1998. Continuous Stenographic Data Transmission Using Uncompressed Audio. Workshop on Information Data Hiding. Portland, Oregon.
17. Lacy, J.; Quackenbush R.; Reibman A.; Shur D.; Snyder J. 1998. On Combining Watermarking with Perceptual Coding. ICASSP Seattle, Washington. MMSP1.9.
18. Lacy, J.; Reibman, A.; Snyder, J. 1997. Watermarking as a Protection Mechanism for IPR in MPEG-4. Fribourg, Switzerland ,ISO/IEC JTC1/SC29/WG11/M2829.
19. Boney L.; Tewfik A. H.; Hamdy K. N. 1996 Digital Watermarks for Audio Signals. EUSIPCO-96, VIII European Signal Proc. Conf., Trieste, Italy. (Patent pending)
20. Bender, W.; Gruhl, D.; Morimoto N.; Lu A. 1996. Techniques for Data Hiding. IBM Systems Journal, Vol. 35, Nos. 3&4.
21. Cox, I. J; Kilian J.; Leithton T.; Shamoon T. 1995. Secure Spread Spectrum Watermarking for Multimedia NEC Research Institute, Technical Report, 95-10.
22. Moskowitz, S. 1998. So This is Convergence? Technical, Economic, Legal, Crypto-graphic, and Philosophical Considerations for Secure Implementations of Digital Watermarking. Available via WWW: <URL: http://www.bluespike.com>.
23. Cox I. J; Kilian J.; Leithton T.; Shamoon T. 1995. Encoding a Hidden Digital Signature onto an Audio Signal Using Psychoacoustic Masking, 7th Int. Conference on Signal Processing Applications & Technology, Boston MA, pp. 476-480, 7-10.
24. Bosi, M.; Brandenburg, K.; Quackenbush, S.; Dietz, M.; Johnston, J.; Herre, J.; Fuchs, H.; Oikawa, Y.; Akagiri, K.; Coleman, M.; Iwadare, M.; Lueck, C.; Gbur, U.; Teichmann, B. 1996 IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC).
25. Yoshida, J. 1998. Watermark Scheme Seeks Contradictory Goals, EETimes, p.50.
26. Yoshida, J. 1998b Digital World Divided on Watermark Specs, EETimes, p. 1.
27. Linnartz, J.; 1998 Fingerprinting and Watermarking. Available via WWW: <URL: http://diva.EECS.Berkeley.EDU:80/~linnartz/>.
28. IFPI 1998. International Federation of the Phonographic Industry Available via WWW: <URL: http://www.ifpi.org>.
29. ISO/IEC JTC1/SC29/WG11 1993. MPEG, International Standard IS 11172-3 Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 Mb/s, Part 3: Audio.
30. Webreference, 1998. Watermarking Companies and Articles. Available via WWW: <URL: http://www.webreference.com/multimedia/watermarks.html>.
31. Zhao, J. 1997. Look its not There - Digital Watermarking. Byte Magazine, January.Available via WWW: <URL:http://www.byte.com/art/9701/sec18/art1.htm>.
32. Tilki, J. F.; Beex, A. A. 1996. Encoding a Hidden Digital Signature onto an Audio Signal Using Psychoacoustic Masking, 7th International Conference on Signal Processing Applications and Technology, Boston MA, pp. 476-480, 7-10.
33. Lathi, B. P. 1998. Modern Digital and Analog Communication Systems, 3rd Edition, Oxford, pp. 179-182.
34. Oppenheim, A. V.; Shaffer, R. W. 1989. Discrete-Time Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
35. Proakis, J. G. 1988. Digital Communications, 3rd Edition, McGraw-Hill, New York
36. Ziemer, R.; Tranter, W. 1990. Principles of Communications, 3rd Edition, Houghton Mifflin, Boston.
37. Kahrs, M.; Brandenburg, K. 1998. Applications of Digital Signal Processing to Audio and Acoustics, Kluwer, Boston.
38. IEEE website: [www.ieee.org](http://www.ieee.org)
39. Google website: [www.google.com](http://www.google.com)

[40] Kamalika Datta, Indranil Sengupta , 2010, “A Redundant Audio Watermarking Technique using Discrete Wavelet Transformation”, Second International Conference on Communication Software and Networks.