Ryerson University
Digital Commons @ Ryerson

Theses and dissertations

1-1-2006

An error resilient scheme of digital watermarking for MP3 streaming audio

Chun Huang
Ryerson University

Follow this and additional works at: http://digitalcommons.ryerson.ca/dissertations

 Part of the Electrical and Computer Engineering Commons

Recommended Citation
Huang, Chun, "An error resilient scheme of digital watermarking for MP3 streaming audio" (2006). *Theses and dissertations.* Paper 463.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

AN ERROR RESILIENT SCHEME of DIGITAL WATERMARKING FOR MP3 STREAMING AUDIO

by

Chun Huang

Bachelor of Engineering, Beijing University of Posts and Telecommunications, Beijing, China, 1994

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2006

© Chun Huang, 2006

UMI Number: EC53857

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform EC53857
Copyright 2009 by ProQuest LLC
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

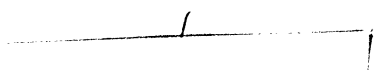
ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.
I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Author's Signature:_

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.



Author's Signature:.

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

An Error Resilient Scheme of Digital Watermarking for MP3 Streaming Audio
Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University
© Chun Huang, 2006

The recent massive growth of networked multimedia systems has caused problems relative to the protection of intellectual property rights. This is particularly true to MP3 audio data. These types of protection systems involve the use of both encryption and authentication techniques. In this thesis we describe a form of authentication known as digital watermarking.

A novel scheme to embed digital watermark to MP3 music using Quantization Index Modulation (QIM) algorithm is proposed in the thesis. A pseudorandom key is generated as watermark message which is embedded into MDCT coefficients of MP3 directly. At the receiver, the MP3 is decoded partially and the watermark is extracted frame by frame. Furthermore we discuss the recovery method of the watermarking after the watermarked MP3 audio is streamed on the network. Distributing digital watermark on Internet through streaming audio is a challenging task because most digital watermarking algorithm is very sensitive to packet loss due to the associated synchronization problem. On the current Internet, packet loss is almost inevitable since its backbone protocols are operating in best-effort manner and do not guarantee the successful delivery of data packets. Therefore it is very essential to

develop a scheme that resists the damage caused by packet loss to audio watermarks. Our robust watermarking scheme can recover the watermarks despite the packets loss (loss rate $\leq 10\%$) on the networks. In addition, we integrate the classical Forward Error Correction (FEC) code in our watermarking scheme, which achieves 100% recovery rate when the packet loss is 6%.

Acknowledgments

I would like to thank and express my gratitude to my advisor Dr. Sridhar Krishnan, whose espertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas. In particular, I want to thank him for teaching me how to write proposals, reports and, of course, this thesis.

A very special thanks goes out to Dr. Bobby Ma, without whose motivation and encouragement, I would not have considered to apply for the M.A.Sc. I appreciate him for his encouragement, guidance and support during my entire graduate study.

I wish also to thank Karthikeyan Umapathy for his continuous help with critical evaluations. I wish to thank him for being available whenever I had problems and for giving endless advice for moving my research forward. I learned a lot from Karthi, Thank You!

I wish to express my gratitude to the members in Signal Analysis Research Group - Jiming Yang, Lam Le, Arunan Ramalingam, April Khademi, Danoush Hosseinzadeh, Lisa Li, for creating such a friendly atmosphere. I will miss the time we spent in the lab.

Last but not least, I would like to thank my wife Xiaoying, for her dedication and understanding, her constant moral and spiritual support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Content distribution on the Internet has become more and more popular and the value of the content being distributed increases, especially for the on-line video and audio Intellectual Properties. In the meantime, the electronic representation and transfer of digitized multimedia information (text, video, and audio) have increased the potential for misuse and theft of such information, and significantly increases the problems associated with enforcing copyrights on multimedia information. Therefore the security of this data has become a main concern for content providers.

Encryption is generally used to safeguard the content while it is being distributed so that unauthorized third-party can not read the stream from the network, but this provides no protection after the intended receiver receives the data. The additional copyright protection can be achieved by watermarking the content [8, 9, 10, 11, 12, 13]. There are two kinds of watermarks: perceptible and imperceptible. For obvious reasons, the latter are more suitable to become part of a digital copyright system. Imperceptible watermarking technologies allow the information owner and provider to secretly embed robust invisible labels in the multimedia material for designating its copyright-related message such as origin, owner, use, content, rights, integrity, or destinations. In order to prevent any copyright forgery, misuse and violation, the embedded copyright labels should be perceptually invisible, unremovable, undetectable, unalterable, and furthermore survives processing which does not seriously

reduce the quality of the multimedia data.

1.2 Problem Statement

One of the critical issues in common watermarking is the synchronization problem. Most existing watermarking schemes apply the spread spectrum technique to embed the watermark signal $w[n]$ into the host audio $x[n]$. The watermarked signal $y[n]$ thus becomes:

$$y[n] = w[n] + x[n]. \tag{1.1}$$

In the receiver side, the intended receiver calculates the correlation $c_{\tilde{w}w}$ between the extracted watermark $\tilde{w}[n]$ and the embedded watermark $w[n]$ as:

$$c_{\tilde{w}w} = \frac{1}{N}\sum_{i=1}^{N} w[i]\tilde{w}[i]. \tag{1.2}$$

The detection can be performed in two possible ways: blind and non-blind detection schemes. Non-blind detection needs both the original audio and the watermarked audio to extract the watermark, while blind case assumes the original audio is unavailable and performs the detection using merely the watermarked audio, which saves the storage space to half. In Non-blind detection, the watermark is extracted by subtracting the original audio $x[n]$ from the received audio $\tilde{y}[n]$ as:

$$\tilde{w}[n] = \tilde{y}[n] - x[n]. \tag{1.3}$$

In blind detection, the correlation in 1.2 is directly measured between the received audio $\tilde{y}[n]$ and the embedded watermark $w[n]$. However, in either blind or non-blind detection, once the estimated watermark $\tilde{w}[n]$ is spatially shifted from the original $w[n]$, the correlation value $c_{\tilde{w}w}$ drops nearly to zero because the watermark signal and its shifted version is designed to be uncorrelated. The spatial shift does not actually remove the watermark from the embedded audio, but the detector fails to find out the presence of watermark any longer. This problem is the so-called synchronization problem.

On the Internet, packet loss is a very common phenomena because the major Internet-working protocols do not guarantee the successful delivery of data packets. These packet loss actually spatial shift the watermarked audio which damages the synchronization of regular watermarks thus rendering the watermark embedding useless. This is true for PCM raw audios. However, for an MPEG-1 Layer III (MP3) audio file, the situation is different in that an MP3 consists of numerous small frames in which the headers are used to delimit and synchronize these frames [14]. One frame loss destroys the data in that frame but does not de-synchronize all the following frames. To take advantage of these frame headers, we embed our watermarks independently in each MP3 frame which synchronize with each individual MP3 frame header. In the case when one packet (one frame) is lost, the following watermarks is not affected because they still synchronize with their own frame headers.

1.3 Motivation and Methods Used

With the rapid development of broadband networks, it is becoming more and more convenient to download digital music from Internet. For many people it is as simple as opening one of many peer-to-peer file share programs like BitTorrent and eDonkey, selecting the tracks, downloading and writing to a media device. Among the numerous audio format on the Internet, MP3 is the most frequently used one because it is not only an open standard that does not need licensing fee but also achieves excellent compression rate against PCM raw audio with CD-like quality retained.

In this thesis, we aim to protect the copyright of MP3 audios that are transmitted on the Internet due to its popularity. Actually watermarking itself will not enhance copyright protection of MP3, but it provides a technological means when combining with digital copyright protection laws [15]. In our watermarking scheme, we embed digital audio watermarks in MP3 compressed domain directly. Because embedding watermarks in MP3 frames instead of PCM data (raw audio) aides us to address the synchronization problem caused by packet loss. Furthermore, we apply Forward Error Correction (FEC) code on the watermarking bits which improves the recovery capacity of the watermarking scheme significantly when

the audio data experience packet loss during the transmission.

1.4 Thesis Organization

Chapter 2 explains the background information of MP3 such as MP3 standard (ISO/IEC 11172-3), MP3 frame structure, and MP3 decoding structures including Huffman decoding, Inverse-quantization and Inverse Modified Discrete Cosine Transform (IMDCT).

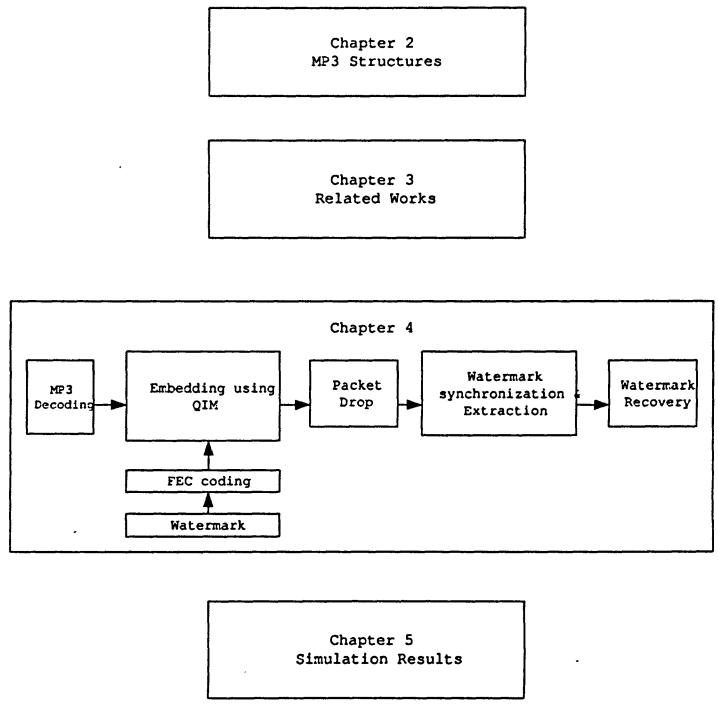


Figure 1.1: Thesis organization

In Chapter 3, we introduce the related works in the literature. Chapter 4 proposes a novel scheme that protects the copyright of MP3 music by embedding the watermarks in MDCT coefficients of MP3 audio using Quantization Index Modulation (QIM) algorithm. In addition, watermarking synchronization is also discussed in this chapter when the MP3 audio experiences packet loss on the networks.

Chapter 5 contains the simulation results. We analyze the packet loss pattern on the Internet and simulate the packet drops to the watermarked MP3 audio. The lost watermarks

are then recovered. The Application Data Unit (ADU) is also introduced to the system to enhance the recovery capacity.

FEC coding is further applied in our recovery scheme. Four types of FEC codes are implemented including Bose-Chaudhuri-Hocquenghem (BCH), Reed-Solomon, Convolutional and Turbo code.

Finally Chapter 6 concludes the thesis by providing a summary of the results of this work and identifying directions for future work.

Chapter 2

Overview of the MP3 standard

2.1 MPEG Standards

The **MPEG-1** standard was developed by the Motion Pictures Expert Group (MPEG) within the International Organization of Standardization (ISO) in November 1992. The audio coding part of the standard (ISO 11172-3) describes a generic coding system, designed to fit the demands of many applications. MPEG-1 Audio consists of three operating modes called "Layers", with increasing complexity and performance, named Layer-1, Layer-2 and Layer-3. Layer-3, with the highest complexity, was designed to provide the highest sound quality at low bit-rates, and is also known as MP3.

The MP3 standard does not exactly specify how the encoding process is to be done. It only outlines the techniques and specifies the format of the encoded audio. By not standardizing the encoder, it leaves the room for evolutionary improvements. The decoder can also be more or less efficiently implemented, depending on the choice of algorithm.

MPEG-2 *[16] denotes the second phase of MPEG. It introduced a lot of new concepts into MPEG video coding, including support for interlaced video signals. The MPEG-2 has higher bitrates (5-10Mbit/s) than MPEG-1. The main application for MPEG-2 is digital television. In 1997, the standard for MPEG-2 Advanced Audio Coding (AAC) was finalized. AAC is a second-generation audio coding scheme for generic coding of stereo and multichannel signals, supporting sampling frequencies from 8 kHz to 96 kHz and a number of audio channels ranging from 1 to 48.*

MPEG-4 [17], with formal as its ISO/IEC designation "ISO/IEC 14496", was finalized in October 1998 and became an International Standard in the beginning of 1999. MPEG-4 builds on the proven success of three fields: digital television, interactive graphics applications (synthetic content) and interactive multimedia (World Wide Web, distribution of and access to content). It provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the three fields. MPEG-4 audio consists of a family of audio coding algorithm - spanning the range from low-rate speech coding (down to 2 kbit/s) up to high-quality audio coding at 86 kbit/s per channel and above. Generic audio coding at medium to high bit-rates is done by AAC.

MPEG-7, formally named "Multimedia Content Description Interface", is a standard for describing the multimedia content data that supports some degree of interpretation of the information meaning, which can be passed onto, or accessed by, a device or a computer code. Unlike MPEG-1, MPEG-2 and MPEG-4, MPEG-7 does not define compression algorithms.

The next addition to the MPEG family of standards is **MPEG-21** [18]. MPEG-21 (ISO/IEC 21000) defines an open framework for multimedia delivery and consumption, with both the content creator and content consumer as focal points. The vision for MPEG-21 is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities.

However if a MPEG-21 system is not designed carefully, the desire to achieve interoperability may be in violation with the requirements to protect the value of the content and the rights of the right holders. Indeed, Digital Rights Management (DRM) systems can go against the very goal of interoperability if they use nonstandardized protection mechanisms. More interoperability in DRM systems is thus crucial to realize an open multimedia infrastructure [19]. To address the DRM issue, MPEG-21 contains two parts in the standardization: Rights Expression Language (REL) and Rights Data Dictionary (RDD) [20]. But these technologies do not solve the problem mentioned in section 1.1.

In this thesis, our focus is on the MPEG-1 Layer III audio.

2.2 Brief overview of the Psychoacoustic Model and Perceptual Coding

The MPEG audio coding standard is based on perceptual audio coding principles. This section gives the introduction to the subject [5, 1, 2, 21].

2.2.1 Psychoacoustic Principles

Human hearing is a magnificent system, with a dynamic range of over 96 dB. However, it is apparent that while we can easily hear a very silent noise like a needle falling, and a very loud noise like an aeroplane taking off, it is impossible to discern the falling needle if we hear the aeroplane at the same time. The hearing system adapts to dynamic variations in the sounds, and these adaption and masking effects form the basics of the psychoacoustic theories and research.

Psychoacoustics is the study of the inner-relation between the ear, the mind, and vibratory audio signal. The field has made significant progress toward characterizing human auditory perception and particularly the time-frequency analysis capabilities of the inner ear. Irrelevant information is identified during signal analysis by incorporating into the encoder some of its principles such as absolute threshold of hearing, critical band frequency analysis, auditory masking, and temporal masking. It takes advantage of the inability of human auditory system to hear quantization noise under conditions of auditory masking. This masking is a perceptual property of the human auditory system that occurs when the presence of strong audio signal makes a temporal or spectral neighborhood of weaker audio signals imperceptible. The results of the psychoacoustic model are utilized in the MDCT block and in the nonuniform quantization block.

Auditory masking consists of three masking principles, which are described below.

The absolute threshold of hearing characterizes the amount of energy needed in a pure tone such that it can be detected by a listener in a silent environment. The absolute threshold is measured in terms of dB Sound Pressure Level (dB SPL). Empirical results also show that the human auditory system has a limited, frequency dependent, resolution. This frequency

dependency can be expressed in terms of critical band widths which are less than 100 Hz for the lowest audible frequencies and more than 4 kHz at the highest. The human auditory system blurs the various signal components within a critical band although this system's frequency selectivity is much finer than a critical band [22]. Twenty-six critical bands covering frequencies of up to 24 kHz have been taken into account.

Frequency masking or simultaneous masking is a frequency domain phenomenon where a low-level signal (the maskee) can be made inaudible (masked) by a simultaneous occurring stronger signal (the masker) as long as masker and maskee are close enough to each other in frequency. Such masking is largest in the critical band in which the masker is located, and it is effective to a lesser degree in neighboring bands [5].

While simultaneous masking is dependent on the relationship between frequencies and their relative volumes, temporal masking may occur when two sounds appear within a small interval of time. Depending on the signal levels, the stronger sound may mask the weaker one, even if the maskee precedes the masker. By placing a sufficient delay between the two sounds the softer sound will be heard. By determining or quantifying the length of time between two tones at which both tones would be audible, temporal masking therefore contributes to data reduction in audio compression. Temporal masking applies both to premasking and postmasking. The duration within which premasking applies is significantly less than that of the postmasking which is in the order of 50 to 200 ms.

Based on the masking phenomenon, a masking threshold can be measured and low level signals below this threshold will not be audible. As Figure 2.1 shows, the masking threshold is escalated in comparison to the curve in noiseless environment. The masked signal can consist of low-level signal contributions, of quantization noise, alias distortion, or of transmission errors. The threshold will vary with time and depend on the sound pressure level, the frequency of the masker, and on characteristics of masker and maskee.

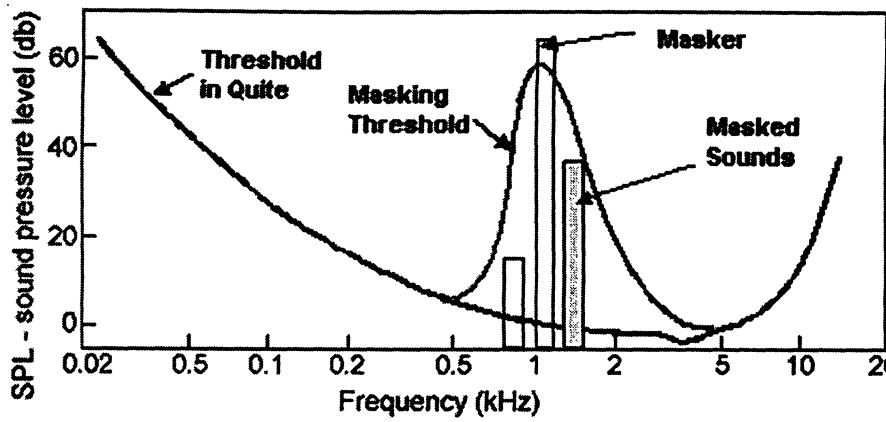


Figure 2.1: Masking threshold

2.2.2 Perceptual Coding

Perceptual coding uses the hypothesis that if the ear cannot perceive some sounds then there is no point in encoding these sounds. Not encoding the sounds that we cannot hear allows a reduction in the overall number of bits needed to encode the signal and therefore the bit rate can be reduced dramatically. This is known as lossy compression coding. Most of the lossy perceptual audio coders follow the general outline of Figure 2.2 as below.

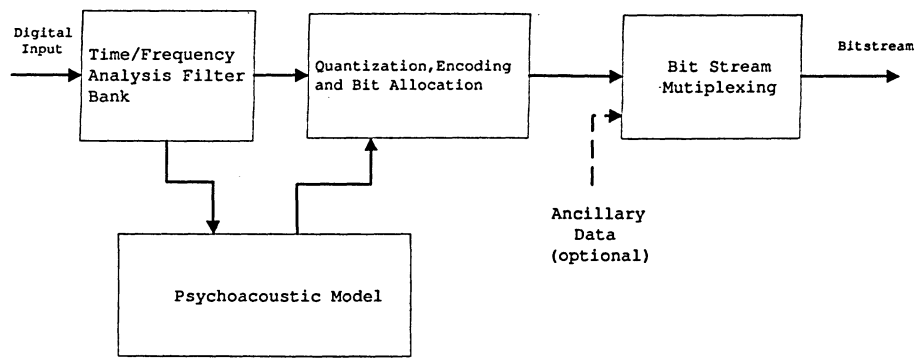


Figure 2.2: Generic perceptual audio coder

The components work as follows:

- The analysis filter bank divides the audio into spectral components. Sufficient fre-

quency resolution must be used in order to exceed the width of the ear's critical bands, which is 100 Hz below 500 Hz and 20% of the center frequency at higher frequencies.

- The psychoacoustic block determines the masking curve, under which noise must fall. It allows the quantization and encoding block to exploit perceptual irrelevancies in the time-frequency parameter set.

- The audio is reduced to a lower bit rate in the quantization and coding section. On the one hand, the quantization must be sufficiently coarse in order not to exceed the target bit rate. On the other hand, the error must be shaped to be under the limits set by the masking curve.

- The quantized values are joined in the bitstream multiplex, along with any side information, and then form the compressed output.

2.3 MP3 Encoding and Decoding Scheme

2.3.1 MP3 Encoding Scheme

The MP3 encoding scheme is depicted as belows:

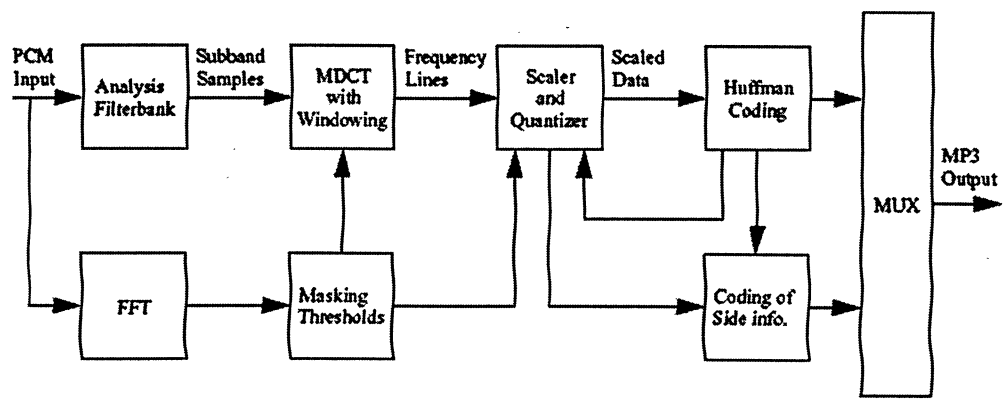


Figure 2.3: *MP3 encoding diagram [1]*

2.3.1.1 Synthesis Polyphase Filterbank

A sequence of 1152 PCM samples, which contains two granules of 576 samples each, are filtered into 32 equally spaced frequency subbands depending on the Nyquist frequency of the PCM signal. If the sampling frequency of the PCM is 44.1 kHz the Nyquist frequency will be 22.05 kHz. Each subband will be approximately $22050/32 \approx 689$ Hz wide. The output of the filters are critically sampled. That means for each granule of 576 samples there are 18 samples output from each of the 32 bandpass filters, which gives a total of 576 subband samples.

2.3.1.2 Modified Discrete Cosine Transform (MDCT)

The subband samples are transformed to the frequency domain through the MDCT. The MDCT is performed on blocks that are windowed and overlapped 50%.

Windowing is done to reduce artifacts caused by the edges of the time-limited signal segment. There are four different window types defined in the MPEG standard. The MDCT is normally performed for 18 samples at a time (long blocks) to achieve good frequency resolution. It can also be performed on 6 samples at a time (short blocks) to achieve better time resolution, and to minimize pre-echoes. There are special windows types for the transition between long and short blocks.

2.3.1.3 Fast Fourier Transform

Simultaneously as the signal is processed by the polyphase filterbank it is also transformed to the frequency domain by a Fast Fourier Transform. A 1024-point FFT is calculated to obtain an estimate of the power density spectrum, which will be used in next block.

2.3.1.4 Masking Threshold

This block retrieves the input data from the FFT output. It uses this input together with psychoacoustic model to determine the masking threshold for all frequencies. This information is useful to decide which window type the MDCT should apply and also to provide the quantization block with information on how to quantize the frequency lines.

2.3.1.5 Scaler and Quantizer

576 spectral values are applied to this block at a time. The masking thresholds are used to iteratively determine how many bits are needed in each critical band to code the samples so that the quantization noise is not audible. This is done iteratively in two nested loops, a distortion control loop (outer loop) and a rate control loop (inner loop).

2.3.1.6 Huffman Coding

The quantized values are Huffman coded. Each division of the frequency spectrum can be coded using different tables so that it achieves better compression rate. Huffman code is a lossless coding system that retains all the information for the quantized bits.

2.3.1.7 Coding of Side Information

All parameters generated by the encoders are collected and stored in the side information part of the frame, such as the Huffman table selection, window switching and scalefactors. It enables the decoder to reproduce the original audio.

2.3.1.8 Bitstream Generation

Finally the bitstream is generated in the last block. The frame header is created followed by side information, CRC (optional), Huffman coded audio data, etc. Each one of these frames

contains 1152 PCM audio samples.

2.3.2 MP3 Decoding Scheme

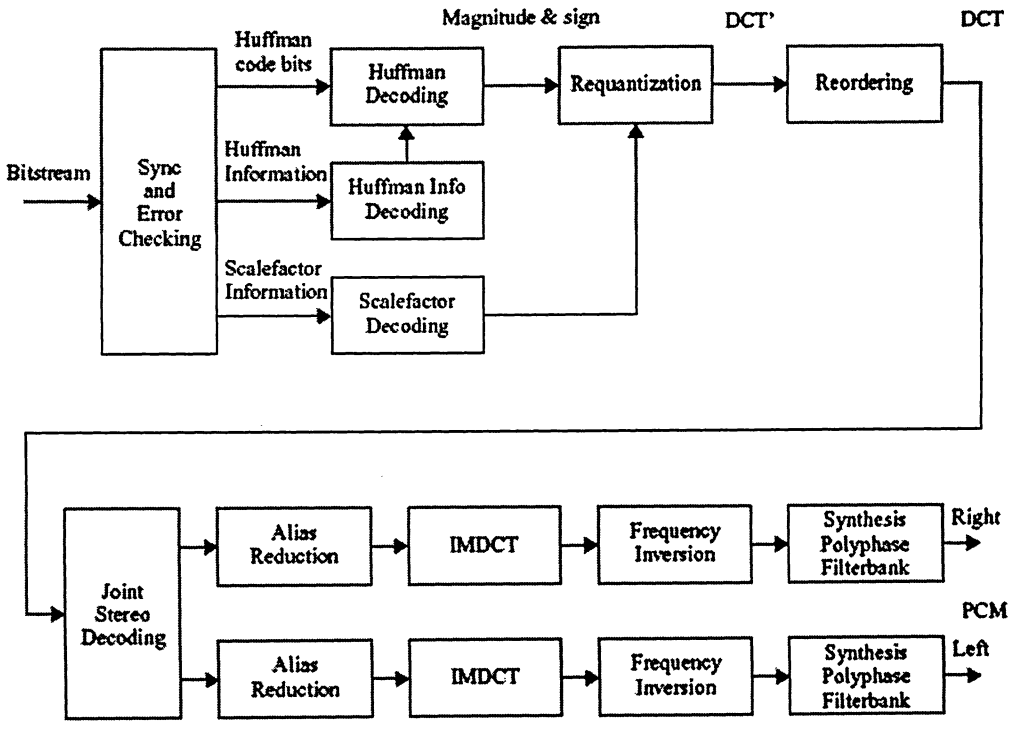


Figure 2.4: MP3 decoding diagram [2]

The MP3 decoding scheme is as shown in Figure 2.4. The compressed MP3 bitstream passes the synchronization and error checking and then the side information is retrieved to store in the buffer for further decoding. The main data passes through a scalefactor decoder to decode scalefactor for inverse quantization. Then the scalefactor and the values from Huffman decoder input the inverse quantizer together to re-build the original spectrum. Finally after the reordering, stereo processing, alias reduction, IMDCT, and synthesis filter bank, the PCM samples are re-generated.

2.3.2.1 MP3 Frame Structure

- Frame Header

  An MP3 audio file is built up of frames. Each MP3 frame consists of frame header, CRC (optional), side information and main data (Figure 2.5). A frame header is constituted by the very first four bytes (32 bits) in a frame, which contains twelve bits "frame sync".

- Side Information

  The side information includes the data that is necessary to decode the main data, such as Huffman table selection, scale factors and window selection. The size of side information is 17 bytes in mono mode and 36 bytes in two channels modes.

- Main Data

  The main data contains the scalefactors and MDCT coefficients (frequency lines) of the audio, the length of which depends on the bit-rate and ancillary data. The main data is processed on blocks that are windowed and overlapped 50%. The main data in a frame consists of two granules with 1152 audio samples totally [14, 23].

- Ancillary Data

  The user-define data is stored in this part. It is not needed for decoding the audio.

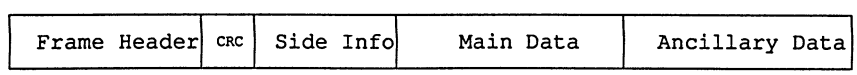


Figure 2.5: MP3 frame format

2.3.2.2 Synchronization and CRC Check

The first step of decoding is to find the synchronization word (0xFFF). Then bit rate, sampling frequency and layer's information that is being used for the encoded audio can be retrieved from the header. If the CRC check bit is set to 1, a sixteen bit CRC code is used for optional error detection.

2.3.2.3 Huffman Decoding

Since the main data contains variable length of Huffman coded samples, a single code word in the middle of the Huffman code bits cannot be identified without starting to decode from a point in the Huffman code bits known to be the start of a code word.

The second task of decoding is to collect data in the side information which describes the characteristics of the Huffman code bits. This includes where to find the Huffman code bits in the bitstream, to decide which Huffman tables are used in each region and whether ESCAPE values are present in the Huffman code bits (Figure 2.6). Moreover, this block must make sure that all frequency lines are generated regardless of how many frequency lines are described in the Huffman code bits. When fewer than 576 frequency lines appear, the Huffman Decoding block must perform zero padding to fill the lack of data.

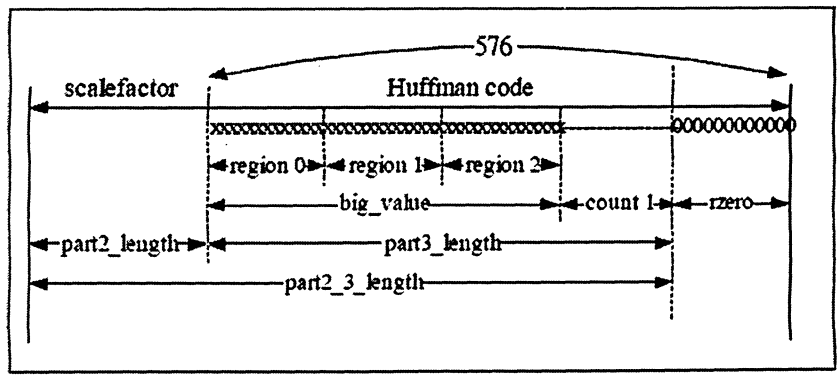


Figure 2.6: Main data organization [3]

2.3.2.4 Scalefactor Decoding

The purpose of the scalefactor decoding block is to decode the coded scalefactors, which is the first section of main data. Input to this block is scalefactor information and coded scalefactors. The output of the block is the decoded scalefactors, to be used in the next requantization block.

2.3.2.5 Requantization

This block conducts the inverse operation of Quantization in encoding diagram. It converts the Huffman decoded data to frequency lines based on the following Equation:

$$xr_i = sgn(iS_i) * |iS_i|^{\frac{4}{3}} * 2^{\frac{1}{4}A} * 2^{-B}. \tag{2.1}$$

The factors A and B in the Equation 2.1 contain different values depending on the block types. For long blocks,

$A = global\_gain[gr] - 210$,

$B = (scalefac\_multiplier * scalefac\_l[gr][ch][sfb]) + (preflag[gr] * pretab[sfb])$;

For short blocks,

$A = global\_gain[gr] - 210 - (8) * subblock\_gain[window][gr]$,

$B = scalefac\_multiplier * scalefac\_s[gr][ch]sfb][window]$,

where $'iS'$ is the quantized value and $'xr'$ is the constructed original value. The *global_gain* is the Quantizer step size information, the $scalefac\_s$ and $scalefac\_l$ are the scale factors for short and long blocks. The $preflag$ is the shortcut for additional high frequency amplification of quantized values. The $scalefac\_multiplier$ equals 0.5 or 1 depending on the value of $scalefac\_scale$ that can be found in side information. The *pretab* is a predefined table in the ISO/IEC 11172-3 standard.

2.3.2.6 Reordering

The frequency lines generated by the Requantization block are not always ordered in the same way. In the MDCT block the use of long windows prior to the transformation, would generate frequency lines ordered first by subband and then by frequency. Using short windows instead, would generate frequency lines ordered first by subband, then by window and at last by frequency. In order to increase the efficiency of the Huffman coding the frequency lines for the short windows case were reordered into subbands first, then frequency and at last by window, since the samples close in frequency are more likely to have similar values. The reordering block will search for short windows in each of the 36 subbands. If short windows are found they are reordered.

2.3.2.7 Stereo Decoding

The MP3 audio supports mono channel and two channels mode. The purpose of this block is to convert the stereo signal to separate left/right signals. The method used for encoding the stereo signal can be read from the mode and mode_extension fields in the header of each frame.

2.3.2.8 Alias Reduction

The alias reduction is required to negate the aliasing effects caused by the overlap of two adjacent overlapped subbands. Only short blocks need to perform the alias reduction. As illustrated in Figure 2.7, the alias reconstruction calculation consists of eight butterfly calculations for each subband, in which the constants cs_i and ca_i are defined in the standard.

2.3.2.9 Inverse MDCT

The IMDCT transforms the frequency lines to polyphase filter subband samples. The analytical expression of the IMDCT is shown in Equation (2.2)

$$x_i = \sum_{k=0}^{\frac{n}{2}-1} X_k \cos[\frac{\pi}{2n}(2i+1+\frac{n}{2})(2k+1)], \quad for\ i = 0\ to\ n-1 \tag{2.2}$$

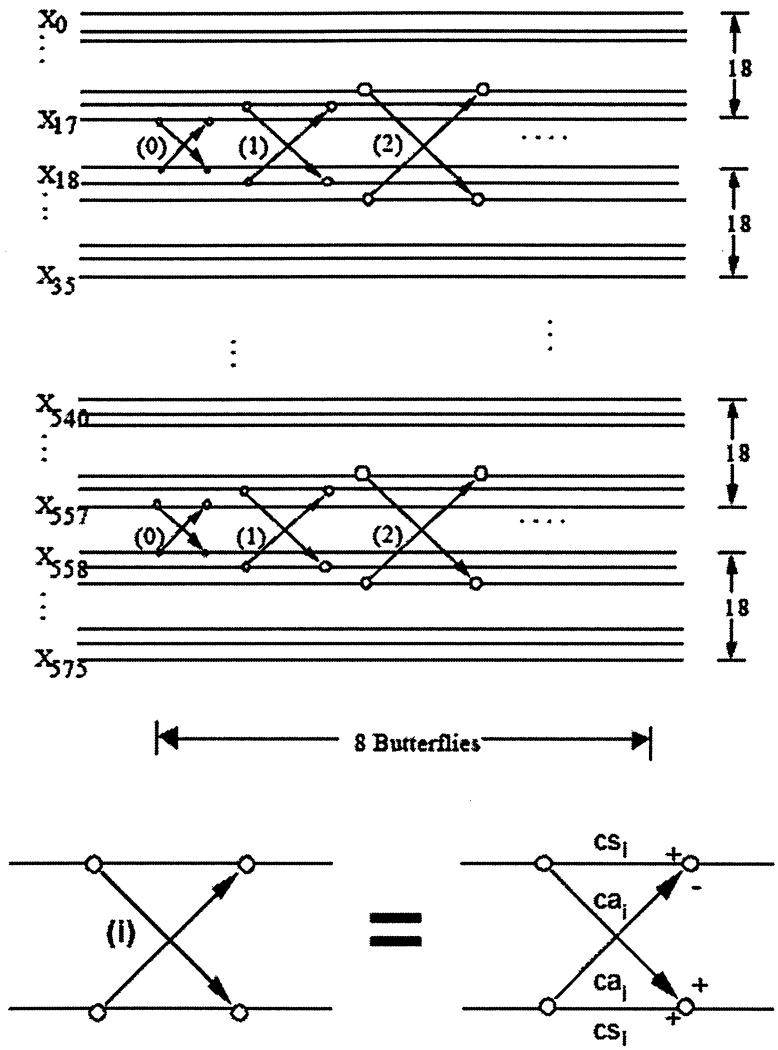


Figure 2.7: Alias reduction butterflies [4]

where:

X_k is the frequency line,

n is 12 for short window and 36 for long window.

2.3.2.10 Frequency Inversion

In order to compensate the frequency inversions in the synthesis polyphase filterbank, every odd time samples of every odd subband is multiplied by -1.

The subbands are numbered [0 - 31] and the time samples in each subband [0 - 17].

2.3.2.11 Synthesis Polyphase Filterbank

The synthesis polyphase filterbank transform the 32 subband blocks of 18 time-domain samples in each granule to 18 blocks of 32 PCM samples. The filterbank operates on 32 samples at a time, one from each subband block, as illustrated in Figure 2.8.

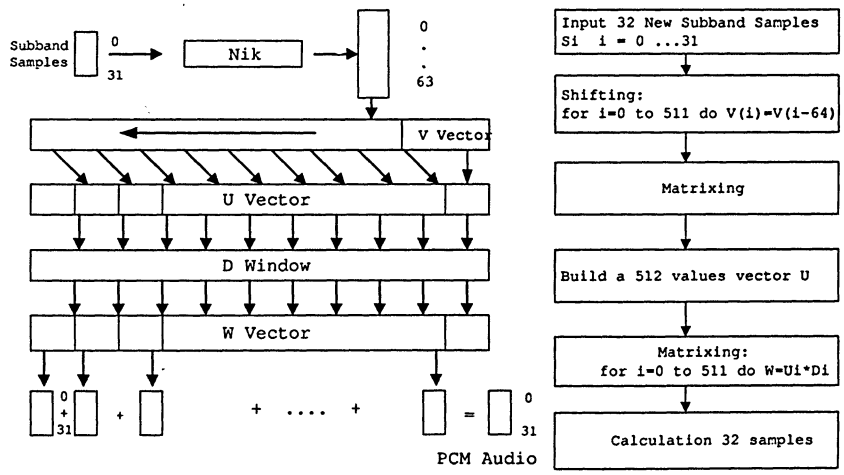


Figure 2.8: Synthesis polyphase filter bank [3]

2.4 Optimized Frame Structure

2.4.1 Background

The most commonly used MP3 audio is constant bit-rate (CBR) and the frame length is calculated as the formula below:

$$FrameLengthInBytes = 144 * BitRate/SampleRate + Padding, \quad (2.3)$$

where padding is 0 or 1.

Therefore the frame length is constant when the bit-rate and sample rate are determined. However, in an audio property, element and features are time-varying in that some audio fragments need less bits to encode and some fragments need more bits to encode. Thus some frames may have extra bits whereas some frames do not have enough bits to encode the corresponding audio data. In MP3 standard, "Bit Reservoir" is used to address this problem. The encoder can donate bits to a reservoir when it needs less than the average number of bits to code a frame. Next, when the encoder needs more than the average number of bits to code a frame, it can borrow bits from reservoir mechanism. The encoder can only borrow bits donated from past frames; it cannot borrow from future frames.

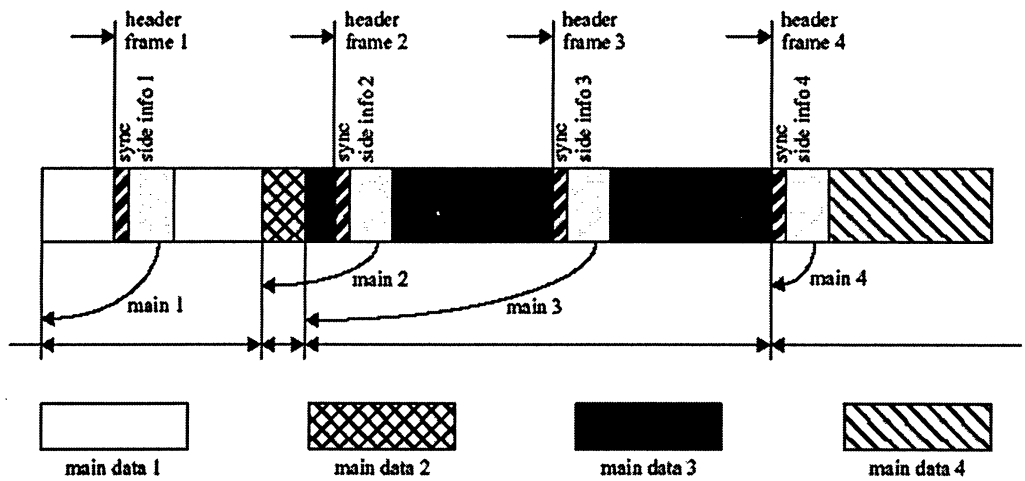


Figure 2.9: Bit reservoir [5]

2.4.2 Application Data Unit (ADU)

The MP3 format depicted in the above paragraph works perfect in storage medias, but it is not optimal for audio streaming since some frames contain a back-pointer to data in earlier frames, and cannot be decoded independently of these earlier frames. Therefore the loss of an MP3 frame will render some data in previous (or future) frames useless, even if they

are received without errors, which degrades the audio quality significantly. To alleviate this problem, a new packetization format is introduced [24]. In this payload format, the MPEG frames are re-arranged so that packet boundaries coincide with "codec frame boundaries" - so called Application Data Unit (ADU). In these new length-variable ADU frames (or packets), no back-pointer is needed and missing one ADU will not affect the neighboring packets. Our watermarking scheme takes advantage of these ADUs by watermarking each ADU instead of the MP3 raw frame. These watermarked ADUs are then distributed to the receivers, where the watermark is detected and the original MP3 frames are formed. Since our watermarks transmit in ADUs, it is advantageous in lossy environment (Figure 2.10).

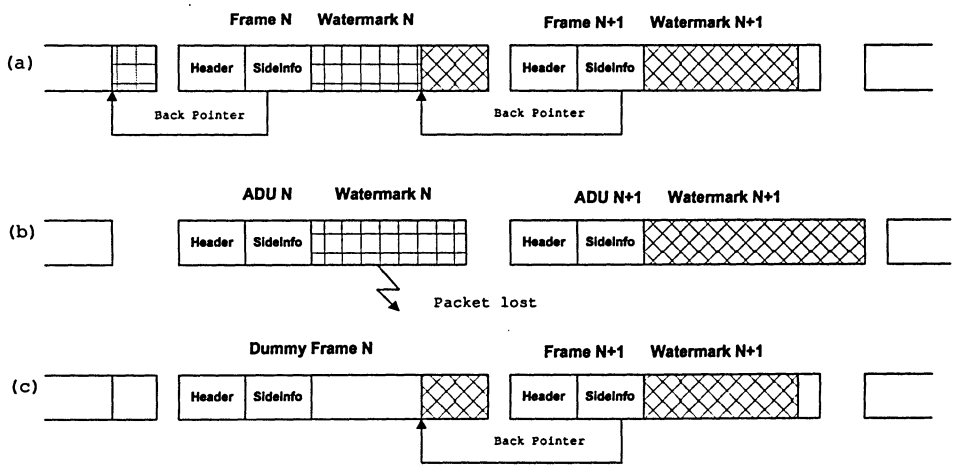


Figure 2.10: (a) Stream of MP3 with watermark (b) Stream of MP3 with ADUs (c) Reconstructed MP3 stream with packet loss

Chapter 3

Review of Previous Works

3.1 MP3 Watermarking Schemes

Many schemes have been proposed in digital audio watermarking research field that claim to be robust against the MP3 compression, which embed watermarks to PCM data (raw audio data) or its transformed domain [25, 26, 27, 28, 29]. Arttameeyanant *et al.* [25] apply psychoacoustic masking and embed a watermark (a color image) in the wavelet transformed domain. Yeo and Kim [26] present a Modified Patchwork Algorithm (MPA), a statistical technique for audio watermarking algorithm in the transform domain. Li *et al.* [27] propose an algorithm of additive audio watermarking based on SNR to determine a scaling parameter α.

Besides these schemes that embed watermarks in PCM or its transformed domain, very few schemes embed watermarks to compressed domain directly. However, audio data are commonly stored and transmitted in compressed format, such as MPEG, when it intends to achieve real-time playability. Thus it would be very beneficial that a watermarking scheme targets to the compressed data domain. Koukopoulos and Stamatiou [30] embed the watermarks in compressed domain by changing the scale factors of the MP3 file during watermark embedding. It uses a crypto mechanism that creates a watermarking key with unique semantic meaning for each user which are robust and cannot be damaged easily as it can survive attacks that achieve to damage a significant percentage of it. The drawback of this scheme is that decoding and re-encoding the MP3 file by a third-party software destroys the

watermarks completely. On the contrary the proposed scheme embeds the watermarks to the Modified Discrete Cosine Transform (MDCT) coefficients which is an integral part of the MP3 audio and it is much more robust against the MP3 re-compression attack.

3.2 Error Resilient Schemes for Multimedia Data on the Networks

When the multimedia data is transmitted over Internet, packets may be lost and the loss tends to occur in burst [31]. This makes the use of error resilience and error concealment necessary. Thus development of simple, robust error-resilient or concealment strategies has become an active research area [32, 33, 34, 35, 36]. R. Parviainen [32] applies a new packetization format for MP3 that provides better error resilience than the standard format. FEC code is applied to further increase the error resilience by reducing the packet loss at the receiver. N. Feamster and H. Balakrishnan [35] propose a scheme that leverage the characteristics of MPEG-4 to selectively retransmit only the most important data in the bitstream. Significant performance gains can be achieved without much additional penalty in terms of latency. When the latency constraints do not permit retransmission, they apply temporal postprocessing at the receiver to recover I-frames that can improve image quality. In addition, J. G. Apostolopoulos [36] presents a system for providing reliable video communication over lossy networks, where the system is composed of two subsystems: multiple state video encoder/decoder and a path diversity transmission system. Multiple state video coding combats the problem of error propagation at the decoder by coding the video into multiple independently decodable streams, each with its own prediction process and state. The path diversity transmission system explicitly sends different subsets of packets over different paths, as opposed to the default scenarios where the packets proceed along a single path, thereby enabling the end-to-end video application to effectively see an average path behavior.

In addition to the error resilient or concealment strategies for multimedia video or audio, there are watermarking techniques that help address the packet loss issues in multimedia

streaming as well [37, 38, 39]. For example, Lin *et al.* [38] propose a method that recovers the lost or corrupted host image information in the lossy environment based on the embedded watermark, which includes the content-based authentication information and recovery information. But to our knowledge, no extensive research has been done to implement the watermarkings in MP3 streaming audio when the audio is transmitted over network and packet drops occur due to transmission errors or network congestion. In our research, we have integrated the watermarking techniques and error correction methods to achieve the best error resilient capacity.

Chapter 4

Approached Scheme

4.1 Audio Watermarking Technologies

In general audio watermarking techniques are classified into five categories: quantization scheme, spread-spectrum scheme, two set scheme, replica scheme, and self-marking scheme [40]. This section gives a brief overview of the state-of-the-art of these watermarking schemes.

4.1.1 Quantization Scheme

A scalar quantization scheme quantizes a sample value x and assign new value to the sample x based on the quantized sample value. In other words, the watermarked sample value y is represented as follows:

$$\begin{aligned} y &= q(x, D) + D/4 \quad if \quad b = 1, \\ y &= q(x, D) - D/4 \quad otherwise, \end{aligned} \tag{4.1}$$

where $q(\cdot)$ is a quantization function, b is a watermarking bit and D is a quantization step. A quantization function $q(x)$ is given as follows:

$q(x, D) = [x/D] \cdot D,$

where $[x]$ rounds to the nearest integer of x. The concept of the simplest quantization scheme in Equation 4.1 is illustrated in Figure 4.1.

The quantizer has a step size of D, and the least significant bit (LSB) is modulated. All reconstruction points marked with a $\times$ have an LSB of 0. Points marked with a $\circ$ have an

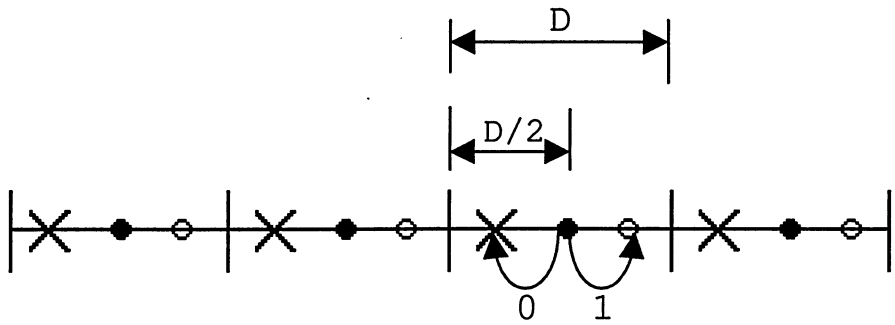


Figure 4.1: A simple low bit modulation quantization scheme

LSB of 1. This process is implemented as this: first quantizing using a quantizer with a step size of D, whose reconstruction points are marked with a $\bullet$, and adding $\pm D/4$.

This scheme is simple to implement. This scheme is robust against noise attack so long as the noise margin is below $D/4$. In other words, the additive noise is larger than $D/4$, then quantized value is perturbed so much that detector misinterprets the watermarking bit.

4.1.2 Spread-Spectrum Scheme

Spread-spectrum watermarking scheme is an example of the correlation method which embeds pseudorandom sequence and detects watermark by calculating correlation between pseudo-random noise sequence and watermarked audio signal. Spread-spectrum scheme is the most popular scheme and has been studied well in literature [11, 41, 42, 43]. The spread-spectrum method can be applied in time domain or transformed frequency domain. Basic idea of this scheme and implementation techniques are described below.

The binary watermark message $w' = \{0, 1\}$ or its equivalent bipolar variable $b = \{1, +1\}$ is modulated by a pseudorandom sequence $r(n)$ generated by means of a secret key K. Each modulated watermark ($w(n) = br(n)$) element w_i is usually called a "chip". Watermarking chips are generated such that they are mutually independent with respect to the original recording x. The watermark $w(n)$ is further scaled according to the required energy of the audio signal $x(n)$. The scaling factor δ controls the trade-off between robustness and inaudibility of the watermark. The modulated watermark $w(n)$ is equal to either $r(n)$ or

$-r(n)$ depending on whether $w' = 1$ or $w' = 0$. The marked signal y is created by:

$$y = x + \delta w, \tag{4.2}$$

The signal variant σ_x^2 directly impacts the security of the scheme: the higher the variance, the more securely information can be hidden in the signal. Similarly, higher δ yields more reliable detection, less security, and potential watermark audibility.

The simplest method of decoding a spread spectrum watermark is the correlation detector, which is optimal for uncorrelated Gaussian host and audio noise. Figure 4.2 shows a diagram of a simple correlation detector. The pseudorandom sequence r is reconstructed from key K, and an estimate of message bit b is calculated from the inner product r and y:

$$\hat{b} = \frac{1}{N}\sum_{i=1}^{N} y_i r_i = E[y \cdot r] + \Gamma(0, \sigma_x/\sqrt{N})^1. \tag{4.3}$$

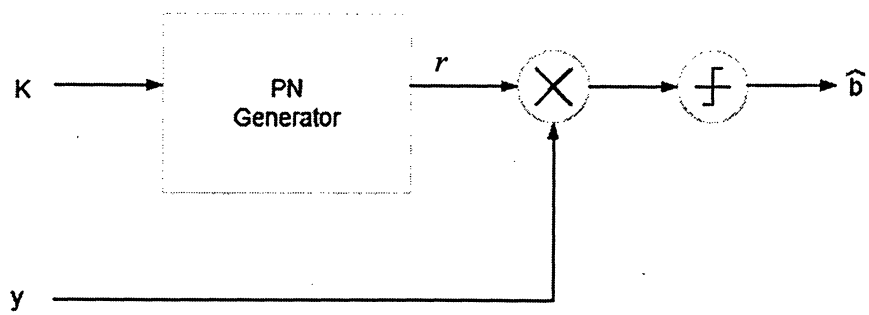


Figure 4.2: Correlation detection of spread-spectrum watermark

Under no malicious attacks or other signal modifications, if the signal y has been marked, the $E[y, r] = \delta$, else $E[y, r] = 0$. $\Gamma(0, \sigma_x/\sqrt{N})$ is a near zero number. The detector decides that a watermark is present if $\hat{b} > \tau$, where τ is a detection threshold that controls the tradeoff between the probabilities of false positive and false negative decisions.

[1] $\Gamma(a, b)$ denotes a Gaussian with mean a and variance b^2

4.1.3 Two Set scheme

Two-set method is a statistical approach for audio watermarking. It pseudo-randomly selects pairs of data and marks them differently with specific statistics depending on the nature of the audio. During decoding process, if two sets are different in the specific pattern, we can conclude that watermark is present. Such decisions are made by hypothesis tests typically based on the difference of means or energy between two sets. Patchwork is one of the popular schemes of two-set method [44, 45].

4.1.3.1 Patchwork scheme

The fundamental patchwork algorithm can be summarized as follows [45].

In the embedding process, two patches are selected pseudorandomly according to a secret key k and then the sample values are modified. A constant value δ is simply added to all values a and subtracted from every b:

$$\begin{aligned} \tilde{a}_i &= a_i + \delta \\ \tilde{b}_i &= b_i - \delta \end{aligned} \tag{4.4}$$

In the detection process, again the secret key k is used to retrieve the two patches. Then, the sum

$$S = \sum_{i=1}^{N} \tilde{a}_i - \tilde{b}_i \tag{4.5}$$

is computed. If the audio actually contained a watermark, we expect the sum to be $2\delta N$, otherwise it should be approximately zero. The detection is based on the statistical assumption

$$E[S_N] = \sum_{i=1}^{N} E[a_i] - E[b_i] = 0, \tag{4.6}$$

which means if we randomly choose several patches in an audio they should be independent and identically distributed. As a consequence, only the owner, who knows the modified locations, can obtain a score close to

$$S = \sum_{i=1}^{N} \{(a_i + \delta) - (b_i - \delta)\} = 2\delta N + \sum_{i=1}^{N} (a_i - b_i) \cong 2\delta N \tag{4.7}$$

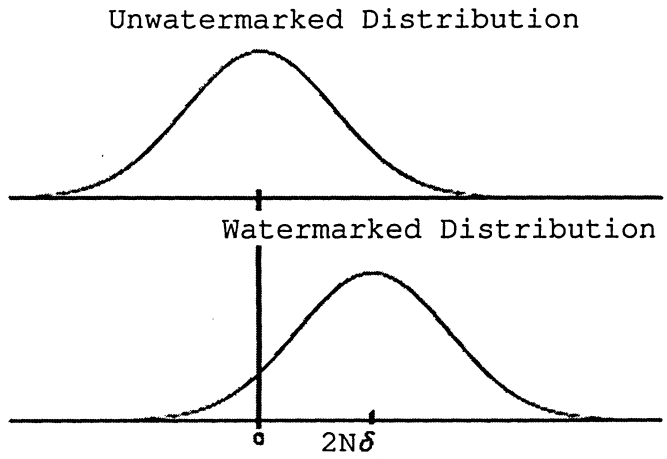


Figure 4.3: A comparison of the unwatermarked and watermarked distributions of the mean difference

The performance of the patchwork scheme depends on the distance between two sample means as shown in Figure 4.3. As N or δ increases, the distribution shifts over to the right. If we shift the watermarked data far enough, any point that is likely to fall into one distribution is highly unlikely to be near the center of the other distribution. It enhances the robustness of the system. However the larger value δ would affect inaudibility.

4.1.4 Replica Scheme

Replica watermarking scheme is robust to synchronization attack because it uses the original audio signal as watermarks which has the same statistical and perceptual characteristics. Echo hiding is one of the popular replica schemes explained in this section.

4.1.4.1 Echo Hiding

Echo hiding is motivated by the fact that human auditory system cannot distinguish an echo from the original when delay and amplitude of the echo are appropriately controlled [46, 47]. After the echo has been added, watermarked signal retains the same statistical and perceptual characteristics.

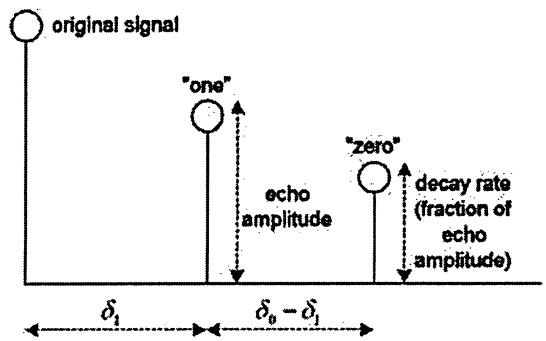


Figure 4.4: Kernels for echo hiding

The fundamental embedding formula is as follows:

$$y(n) = x(n) + \alpha x(n - \delta), \quad \delta = \delta_1 \; or \; \delta_0 \tag{4.8}$$

The watermarking embedding process can be represented as a system that has one of two possible functions. In the time domain, the system function are discrete time exponentials, differing only in the delay between impulses. Processing host signal through any kernel in Figure 4.4 will result in an encoded signal. The delay between the original signal and the echo is dependent on the kernel being used, 1 if the "one" kernel is used and 0 if "zero" kernel is used.

The extraction of the embedded information involves the detection of delay δ. The magnitude of the autocorrelation of the encoded signal's cepstrum

$$
\begin{aligned}
C &= F^{-1}\left\{log(|F(x)|^2)\right\} \\
w &= 1 \quad if\ C_1 > C_0 \\
w &= 0 \quad otherwise
\end{aligned}
\tag{4.9}
$$

where F represents the Fourier Transform and F^{-1} the inverse Fourier Transform. w is the embedded watermark. C_1 is the autocepstrum at location δ_1 and C_0 is the autocepstrum at location δ_0.

Increased robustness of the watermark algorithm requires high-energy echoes to be embedded which increases audible distortion. There are several modification to the basic echo-hiding algorithm that enhance robustness or impercepbility. Double echo [48] is one of them.

$$
y(n) = x(n) + \alpha x(n-\delta) + \alpha x(n+\delta) \tag{4.10}
$$

This modified echo hiding scheme introduces both post-echo and pre-echo to the original audio. The virtual pre-echo plus the regular post-echo makes the cepstrum peak higher than single echo with same strength of echo α. Thus it increases the detection rate with larger cepstrum peak or increase imperceptibility by reducing α accordingly.

4.1.5 Self-marking Scheme

Self-marking method embeds watermark by leaving self-evident marks into the signal [40]. This method embeds special signal into the audio, or change signal shapes in time domain or frequency domain. Time-scale modification method [49, 50] and many schemes based on the content analysis (salient features) [51] belong to this category.

4.1.5.1 Time-Scale Modification

Time-scale modification refers to changing the time duration of an audio sample without changing the pitch or other spectral characteristics [50]. Because the pitch is not changed,

small amount of time scale modification are typically not noticeable. As shown in Figure 4.5, short time regions of the signal are either compressed or expanded by an imperceptible amount.

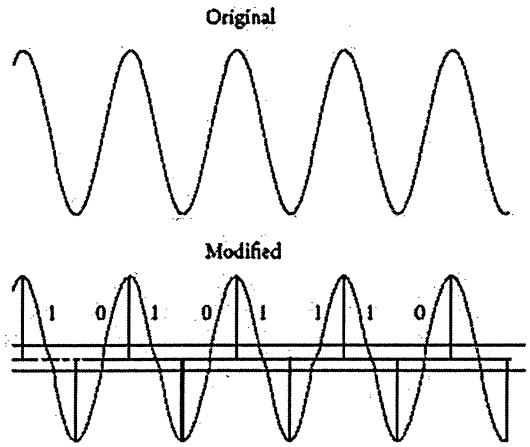


Figure 4.5: Simple sample of time-scale method

In the embedding process, the envelope of the input signal is calculated from which a predefined value and extrema are picked. The extremas are then refined so that only strong ones are preserved. The length of the intervals between successive extrema are quantized and the quantization index is forced to be either odd or even to embed one or zero [49, 52]. During extraction procedure, watermark data is recovered by detecting whether the regions are an odd or even length.

4.1.5.2 Content Analysis

Basic idea of Content analysis [51] is to extract salient point as locations where the audio signal energy is climbing fast to a peak value. These points are special and noticeable signal to the embedder, but common signal to the attackers. This scheme performs content analysis using the wavelet filterbank while the watermark is embedded in the Fourier transform domain. It intentionally modifies signal shape to keep salient points, which is sufficiently invariant under malicious modifications. The salient features can be used especially for

synchronization or for robust watermarking, for example, against time-scale modification attack.

4.2 Quantization Index Modulation (QIM)

4.2.1 QIM Algorithm

QIM is an optimized quantization embedding algorithm originally proposed by B. Chen and G.W. Wornell [6, 53]. We choose QIM as our embedding algorithm because QIM systems in general offer significant performance advantages over previously proposed spread-spectrum and low-bit modulation systems in terms of the achievable trade-offs among information-embedding rate, distortion, and robustness. QIM refers to embedding information by first modulating an index or sequence of indices with the embedded information and then quantizing the host signal with the associated quantizer or sequence of quantizers.

Figure 4.6 illustrates this QIM information-embedding technique. In this example, one bit is to be embedded so that the information $m \in \{1, 2\}$. Thus, we require two quantizers, and their corresponding sets of reconstruction points in $\Re^N$ (host signal $\mathrm{x} \in \Re^N$) are represented in Figure 4.6 with O's and X's. If $m = 1$, the host signal is quantized with the X-quantizer, i.e., composite signal value s ($s \in \Re^N$) is chosen to be the closest to X. If $m = 2$, X is quantized with the O-quantizer.

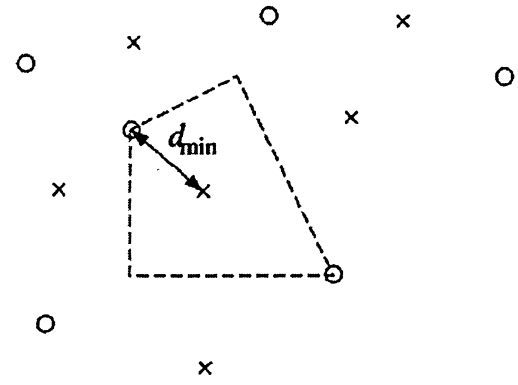


Figure 4.6: QIM for information embedding [6]

As x varies, the composite signal value s varies from one X-point ($m = 1$) to another or from one O-point to another ($m = 2$), but it never varies between a X point and a O point. Thus, even with an infinite energy host signal, one can determine if channel perturbations are not too severe. The X points and O points are both quantizer reconstruction points and signal constellation points, and we may view design of QIM systems as the simultaneous design of an ensemble of source codes (quantizers) and channel codes (signal constellations).

In fact, properties of the quantizer ensemble can be related directly to the performance parameters of rate, distortion, and robustness. For example, the number of quantizers in the ensemble determines the information-embedding rate R. The sizes and shapes of the quantization cells determine the embedding-induced distortion, all of which arises from quantization error. For many classes of channels, the minimum distance

$$d_{min} \triangleq \min_{(i,j):i\neq j} \min_{(x_i,x_j)} ||y(x_i;i) - y(x_j;j)|| \tag{4.11}$$

between the sets of reconstruction points of different quantizers in the ensemble effectively determines the robustness of the embedding [54].

As in [6, 53], an implementation of QIM is dither quantizers, which have the property that the quantization cells and reconstruction points of any given quantizer in the ensemble are shifted versions of the quantization cells and reconstruction points of any other quantizer in the ensemble. The host signal is quantized with the resulting dithered quantizer to form the composite signal. Specially, we start with some base quantizer $q(.)$, and the embedding function is

$$y(x;m) = q(x + d(m)) - d(m), \tag{4.12}$$

where $d(m)$ is a pseudorandom vector called dither vector.

Intuitively, the minimum distance measures the size of perturbation vectors that can be tolerated by the system. In the case of the bounded perturbation channel[2], the energy bound

[2]In bounded perturbation channel, we consider the largest perturbation energy per dimension σ_n^2 and every perturbation vector satisfies $||n||^2 \leq N\sigma_n^2$

of $N\sigma_n^2$ implies that a minimum distance decoder is guaranteed not to make an error as long as

$$\frac{d_{min}^2}{4N\sigma_n^2} > 1. \tag{4.13}$$

If the channel is an Additive White Gaussian Noise (AWGN) channel with a noise variance of σ_n^2, at high SNR the d_{min}^2 also reflects the error probability of the minimum distance decoder,

$$P = Q\left\{\sqrt{\frac{d_{min}^2}{4N\sigma_n^2}}\right\}. \tag{4.14}$$

Assuming the quantization steps are sufficiently small such that x can be modeled as uniformly distributed within each cell, and the dither vector $d_i \subset \{-\Delta/2, +\Delta/2\}$, the expected squared-error distortion per sample is

$$E[D(y,x)] = E[(q(x+d(m)) - d(m) - x)^2] = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} z^2 dz = \frac{\Delta^2}{12}, \tag{4.15}$$

4.2.2 Spread Spectrum and Spread Transform Dither Modulation

A large number of spread spectrum watermarking algorithms have been proposed in the literature and the simplest embedding function can be expressed in Equation 4.2. For this class of embedding methods, the host signal x acts as additive interference that inhibits the decoder's ability to estimate w. Consequently, even in the absence of any channel perturbations ($n = 0$), one can usually embed only a small amount of information. Thus, these methods are useful primarily when either the host signal is available at the decoder or when the host interference is much smaller than the channel interference [54].

To enhance the capability of the Dither Modulation QIM system, a Spread Transform Dither Modulation (STDM) method is proposed [6]. The STDM is the combination of spread spectrum and dither modulation in that one can convert a spread-spectrum of the Equation (4.2) into a STDM system by replacing addition with quantization,

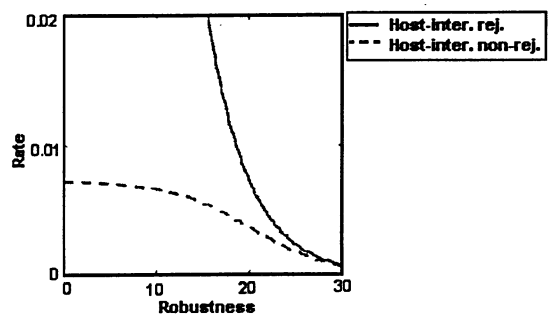


Figure 4.7: Qualitive behavior of host-interference rejecting and non-rejecting embedding methods

$$\tilde{y} = y^T v = q(\tilde{x} + d(m)) - d(m), \tag{4.16}$$

where $\tilde{x}$ is the projection of the host signal, $\tilde{x} = x^T v$. v is the spreading vector.

Actually much efforts have been invested into spread spectrum method to enhance the robustness and rate, *e.g.*, taking advantage of Human Auditory System (HAS) or Human Visual System, applying perceptual masking, exploiting joint time-frequency characteristics, etc. An additional advantage of STDM specifically over other forms of dither modulation is that one can easily convert existing additive spread spectrum into STDM system while retaining the other optimized components of the system.

4.2.3 SNR Advantage of STDM

In this section we focus our analysis on the representative case of embedding one bit in a length-L block of x using an unit energy spreading vector v. In the embedding function of SS system as Equation (4.2), the δ is a scalar function of the message x, w can be seen as the spreading vector containing the watermarking information. The squared-error distortion is

$$D(y,x) = \frac{1}{N}||y - x||^2, \tag{4.17}$$

and its expectation is $D_y = E[D(y,x)]$. Thus $\delta(0) = \pm\sqrt{LD_y}$ in Equation (4.2) and

$$|\delta(0) - \delta(1)|^2 = 4LD_y. \tag{4.18}$$

For STDM (4.16)

$$\min_{(\tilde{x_1},\tilde{x_2})} |\tilde{y}(\tilde{x_1}, 1) - \tilde{y}(\tilde{x_2}, 2)|^2 = \frac{\Delta^2}{4} = 3LD_y, \tag{4.19}$$

where $\Delta^2 = 12LD_y$ based on Equation 4.15.

The decoder makes the decision based on the received signal $\tilde{s}$, the projection of the channel output s onto v. In the case of additive spread spectrum, $\tilde{s} = \delta(x) + \tilde{x} + \tilde{n}$, while in the case of STDM, $\tilde{s} = \tilde{y} + \tilde{n}$, where $\tilde{n}$ is any kind of "'noise"' either through normal signal operation or by intended attack. We let $P(\cdot)$ be some measure of energy. The energy of the interference is $P(\tilde{x}+\tilde{n})$ for additive spread spectrum, but only $P(\tilde{n})$ for STDM. So the SNR for SS system is

$$SNR_{ss} = \frac{4LD_y}{P(\tilde{x}+\tilde{n})}, \tag{4.20}$$

and for STDM is

$$SNR_{stdm} = \frac{3LD_y}{P(\tilde{n})}. \tag{4.21}$$

Thus the advantage of STDM over SS is

$$\frac{SNR_{stdm}}{SNR_{ss}} = \frac{3}{4}\frac{P(\tilde{x}+\tilde{n})}{P(\tilde{n})}, \tag{4.22}$$

which is typically very large since the noise $\tilde{n}$ is usually much smaller than the host signal x otherwise the signal quality will be noticeably affected.

4.2.4 Other Improvements to QIM System

However there is a major drawback of the QIM scheme in that it is sensitive to amplitude scaling attack because the step size Δ is usually small in order to keep the quality of the watermarked signal. If the noise energy $\sigma_n^2 > \sigma_w^2$, the system is not robust. Several methods

have been proposed in the literature to enhance the robustness and performance of the QIM system [55, 56].

Eggers *et al.* developed a Scalar Costa Scheme (SCS) that significantly increased the performance of the QIM. The embedding and detection algorithms are as follows.

(1). Embedding using Scalar Uniform Quantization

First a random codebook must be designed

$$U^N = \left\{ \tilde{u}_l = \tilde{w}_l + \alpha \tilde{x}_l | l \in \{1, 2, ..., L\}, \tilde{w} \sim N(0, \sigma_w^2 I_N), \tilde{x} \sim N(0, \sigma_x^2 I_N) \right\} , \tag{4.23}$$

where $\tilde{w}$ and $\tilde{x}$ are independent, L is the size of the codebook, $I(u; y)$ is the mutual information between the codebook entries and the received signals, and N is the codeword length. Practically one-dimensional codebook is used in the embedding

$$U^1 = \left\{ u = k\alpha\Delta + d\frac{\alpha\Delta}{2} | d \in \{0, 1\}, k \in Z \right\} . \tag{4.24}$$

Then the codebook is scaled and quantized as $Q(x, \frac{U^1}{\alpha})$ and finally the watermark is generated

$$\tilde{w} = \tilde{u} - \alpha \tilde{x}. \tag{4.25}$$

The embedding scheme is depicted in Figure 4.8

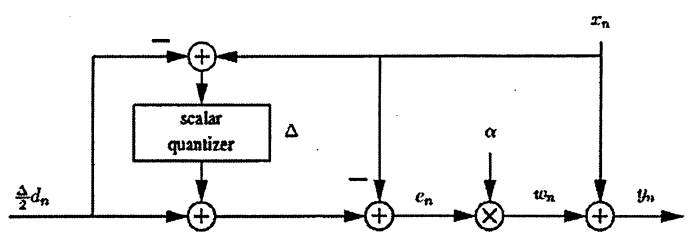


Figure 4.8: Embedding diagram of SCS

(2). Detection Based on Scalar Uniform Quantization

The detection scheme of the SCS is similar with QIM system. The decoder has the access to the same codebook. Treating this codebook as a quantizer, the decoder acts as if it quantizes the received data $y' = x + w + v$ to the closest codebook entry. The watermark will be successfully retrieved if the y' falls into the correctly indexed quantization bin.

In the SCS system the α is optimized for each WNR to achieve the best transmission performance. For a given watermark power σ_w^2, the α is determined as

$$\alpha = \sqrt{\frac{\sigma_w^2}{E\{e^2\}}} = \sqrt{\frac{12\sigma_w^2}{\Delta^2}}, \tag{4.26}$$

Where e is the quantization error. The Dither Modulation can be treated as a special case of SCS where $\alpha = 1$.

4.2.5 Proposed Watermark Embedding Scheme

As mentioned in Chapter 2, the perceptual masking is applied to the audio data during the encoding process of the MP3. Thus MP3 can achieve good compression ratio while still retaining sound audio quality. Figure 4.9 shows the difference of the audio in time-frequency domain after it is processed in MP3 channel. Although this change can hardly be differentiated by human auditory system, it actually introduces large amount of noise to watermarking system.

The MDCT coefficients are obtained after the perceptual shaping to the input signal. These coefficients are very invariant because they represent the perceptually most significant components of a specific audio. They are supposed to be relatively stable under common signal processing manipulation such as low-pass filtering, equalization, noise addition, echo addition and resampling. Thus we choose these coefficients as our watermarking embedding region. As shown in Figure 4.10, after we have finished the alias reduction in the MP3 decoding process, we embed the watermarks into the MDCT coefficients. Each MP3 frame has two granules, each one of which consists of 576 coefficients, representing the frequency of 0 - 22 kHz. We adjust the lowest 300 MDCT coefficients and hide the watermarks there. During the decoding process the watermark can be extracted without fully decoding the MP3

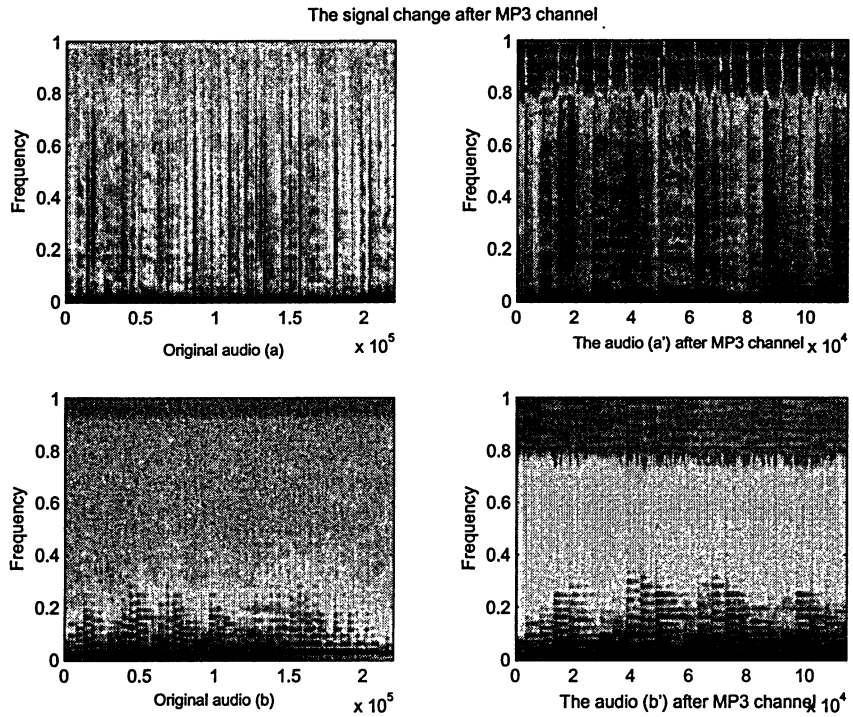


Figure 4.9: The frequency element change of the audio after MP3 channel

music. By doing this we also save major computational time because research shows that steps of Inverse MDCT (IMDCT), Frequency Inverse and Synthesis Filter Bank in Figure 4.10 consume 90% of total MP3 decoding time (Table 4.1).

To embed the watermarks to the audio signals, we first pseudorandomly generate a length-L vector $d_i(1) \subset \{-\Delta/2, +\Delta/2\}$, which represents the information bit $b_0 \in \{0, 1\}$. Then we generate another vector,

$$d_i(2) = \begin{cases} d_i(1)+\Delta/2, & d_i(1)<0 \\ d_i(1)-\Delta/2, & d_i(1)\geq 0, \end{cases} \quad i = 1, ..., L\,, \tag{4.27}$$

which represents the information bit b_1. Each element of vectors $d_i(1)$ and $d_i(2)$ have the distance as far as $\Delta/2$ so that it makes the embedding system more robust. After the two vectors are created, we use the uniform quantizer with step size Δ and encode the watermarks as in Equation 4.12.

The embedding algorithm is depicted in Figure 4.11. FEC code is further applied in the

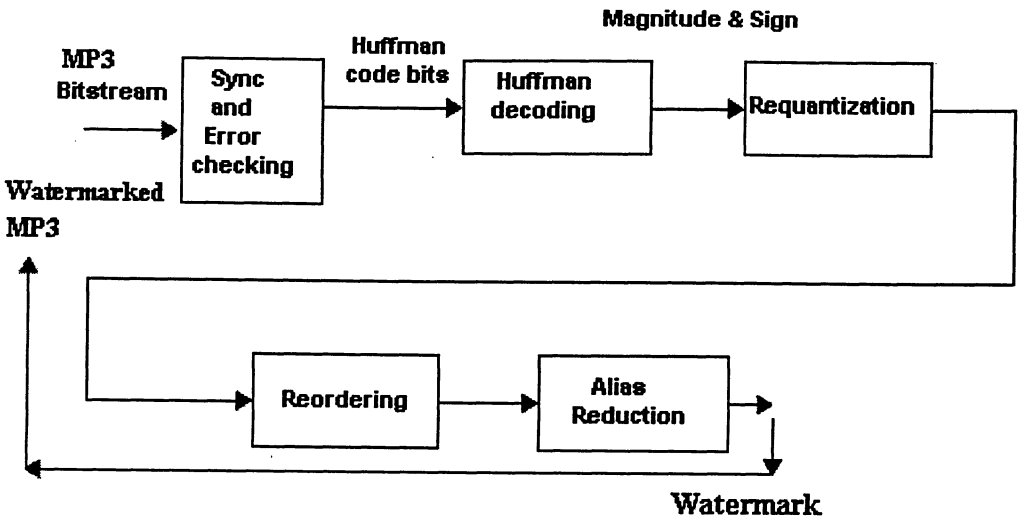


Figure 4.10: Embedding into MP3

scheme and the detailed information is explained in Section 4.4.

In this scenario, the reconstructed points are shifted by $\pm\Delta/2$ in each dimension relative to the points of any other quantizer over at least Ld_H dimensions. Thus the minimum distance squared (Figure 4.6) is

$$d^2_{min} = d_H \frac{k_u}{k_c} L * (\frac{\Delta}{2})^2, \tag{4.28}$$

where d_H is the minimum Hamming distance, $\frac{k_u}{k_c}$ is the code rate for error correction code

	Music1	Music2	Music3
Side-info Dec.	0.09%	0.08%	0.08%
Scale-fac Dec.	0.15%	0.27%	0.31%
Inv. Quant.+Huffman Dec.	8.17%	8.95%	8.38%
Stereo Dec.	0.01%	0.02%	0.00%
Reordering	0.57%	0.68%	0.79%
Alias Red.	0.68%	0.75%	0.53%
IMDCT+Fre-inv.	15.27%	15.33%	14.98%
Syn. Filterbank	75.06%	73.90%	74.93%

Table 4.1: Decoding time at different MP3 decoding phase

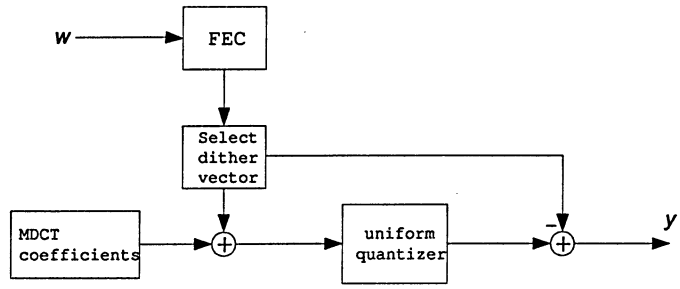


Figure 4.11: Proposed embedding algorithm

(in the uncoded case, $\frac{k_u}{k_c}$=1).

In our scenario, if we do not consider the malicious attacks to the watermarked audio, the normal processing of MP3 audio itself introduces the most significant distortion to the watermarks, especially in the quantization/requantization step in Figure 4.10. The Equation 4.29 is the formula for quantization in encoding process, which shows that the amount of quantization distortion is dependent on the parameters of global_gain and scale_factors. These factors are varied with different type of audios.

$$iS_i = sgn(xr_i) * (xr_i * 2^B * 2^{-\frac{1}{4}A})^{\frac{3}{4}}, \tag{4.29}$$

where $A = global\_gain[gr] - 210, B = (scalefac\_multiplier * scalefac\_l[gr][ch][sfb]) + (preflag[gr] * pretab[sfb])$. The *global_gain* is the Quantizer step size information, the $scalefac\_s$ and $scalefac\_l$ are the scale factors for short and long blocks. The *preflag* is the shortcut for additional high frequency amplification of quantized values. The *pretab* is a predefined table in the ISO/IEC 11172-3 standard (Section 2.3).

In our simulations, the watermark to quantization noise is:

$WNR = \sum(W_s - X)^2 / \sum(W_s - Q_{ws})^2 = 5.7$ dB,

where W_s is the watermarked signal, X is the original signal, and Q_{ws} is the requantized watermarked signal.

4.3 Distributing Watermarks on the Networks

When an mp3 file is transmitted in the Internet, some packets may be lost [57]. The survey of [58] shows that it is inevitable that some receivers will experience packet loss in a large multicast conference. Actually packet loss in an IP network occurs as a result of congestion in the network [59]. As more packets are placed on the network the finite length buffers at network nodes can rapidly become full causing packet to be discarded or lost. Networking protocols such as TCP retransmit any lost packets but this is time consuming and not suitable for real-time application. The occurrence of packet loss on Internet is burst-like in nature and generally loss rate is less than 2% - 5%.

4.3.1 Watermark Synchronization

At the receiver's side not all frames will be received due to the packet loss mentioned in the previous section. Usually this damages the watermarks embedded in the audio since it spatially shifts the audio samples which breaks the synchronization (or correlation) between the received audio and watermarking keys (or vectors) (Figure 4.12). Since the synchronization problem has been such a critical issue for audio watermarking, numerous schemes have been proposed in the literature to address this issue.

4.3.1.1 Feature Extraction Methods

An audio can be represented by multiple features such as energy, zero-crossing rate, pitch, beat, and frequency centroid, and the combination of some of the features are unique like the fingerprint of a person. Developing the watermarking schemes that take advantage of the features invariant under attacks to serve as synchronization point have been attracting the attention of many researchers in the literature [51, 60, 61, 62, 63].

As mentioned in section 4.1.5.2, Wu *et al.* [51] proposed a salient point definition which satisfies the requirement of synchronization tag. A salient point is the energy fast climbing part of audio signals. Hsieh *et al.* [61] improved the salient point method by checking the energy of cepstrum coefficients and embedding the blind watermark in cepstrum domain.

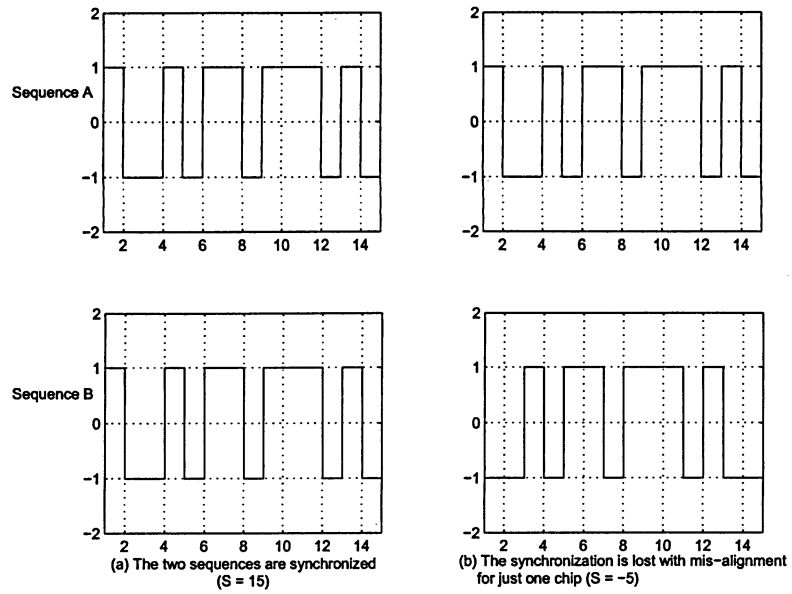


Figure 4.12: (a)Sequences A and B are synchronized with $S = \sum A \cdot B = 15$; (b)The first chip of B is missing so A and B are de-synchronized with $S = \sum A \cdot B = -5$.

Li *et al.* [62] designed a robust scheme based on music edge detection. The basic embedding strategy is (1) all envelope peaks of the original audio are calculated, corresponding local regions are selected as embedding regions; (2) FFT is performed on each region, AC FFT coefficients from 1 to 6 kHz are selected as the dataset for watermark embedding; (3) each watermark bit is repeatedly embedded into all the selected embedding regions by exchanging AC FFT coefficient pairs. The same method is used to detect the watermark.

Besides the synchronization schemes that check the steady features of the audio, novel methods based on statistical feature manipulation are also proposed in the literature. In the paper [60], Li *et al.* adopt the mean of the coefficients value at the coarsest approximation subband of wavelet decomposition as the statistical feature. This statistical feature is relatively invariant under normal signal processing. The embedding algorithm in [60] is

1. the input audio signal is first segmented into overlapped frames and then is Hamming-windowed by $w(i) = 0.54 - 0.46 * \cos(2\pi i/256)$ to minimize the Gibbs effects;

2. for each audio frame, five level wavelet decomposition is performed with the 9/7 bi-

orthogonal wavelet basis and the mean of all the wavelet coefficients at the coarsest approximation subband is calculated;

3. the mean is removed from all the coefficients. Then the watermark bit is embedded by adjusting the coefficients value with $\pm\Delta$ depending on the message is "1" or "-1".

The detection method for Li *et al.*'s scheme is straightforward. The mean of the wavelet coefficients at the coarsest approximation subband of 5-level wavelet decomposition is calculated. If it is larger than zero, the bit "1" is extracted; otherwise "0" is extracted.

4.3.1.2 Synchronization Code and Block Repetition Coding

In addition to the feature extraction methods, some schemes achieve the synchronization without relying on the feature of the audio.

A robust scheme using synchronization code in time domain is developed by Huang *et al* [64]. The synchronization code should have a high-peak auto-correlation when it is matched while involving few data bits and low computation complexity to achieve robustness and inaudibility. Bark code with 12 bits (for example 111110011010) is adopted in their paper. To embed the Bark code successfully, this method sets the lowest 13 bits to be "1100000000000" when embedding "1", to be "0100000000000" when embedding "0". After the synchronization code is placed, the watermark message is embedded thereafter.

HAS is much more tolerable to constant scaling rather than variations in scaling over time. Based on this fact, Kirovski *et al* present a block repetition coding of chips method to resolve the synchronization issue [65]. The basic idea of the method is shown in Figure 4.13, in which the correlation is not broken if the chips are shifted not so severely. Because at the receiver side, only the central sample of each expanded chip is used for computing correlation (correlation is calculated at the areas with solid vertical line only). So by using such a repetition chip encoding with expansion by R chips, correct detection is available up to $R/2$ chips off misalignment. Of course this method enhances robustness at the cost of embedding capacity [40].

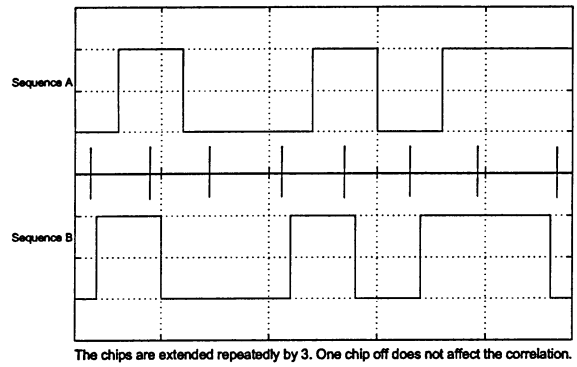


Figure 4.13: The concept of block repetition coding of chips

4.3.1.3 Our Implicit Synchronization Method

Comparing with the above methods that are specially designed to address the synchronization issue, our watermarking scheme is robust against the packet-loss-induced synchronization problem by nature and there is no extra computation overhead. It can be seen as implicit synchronization scheme. Because the watermarking bits are self-synchronized frame by frame with the frame header in our scheme. Thus as long as an MP3 frame (codec frame) is successfully received, the watermarking bits contained in the frame can be successfully recovered (Figure 4.14).

Our scheme can also integrate with other watermarking methods to enhance robustness. For example, applying the synchronization code in Section 4.3.1.2 before embedding the watermark using QIM.

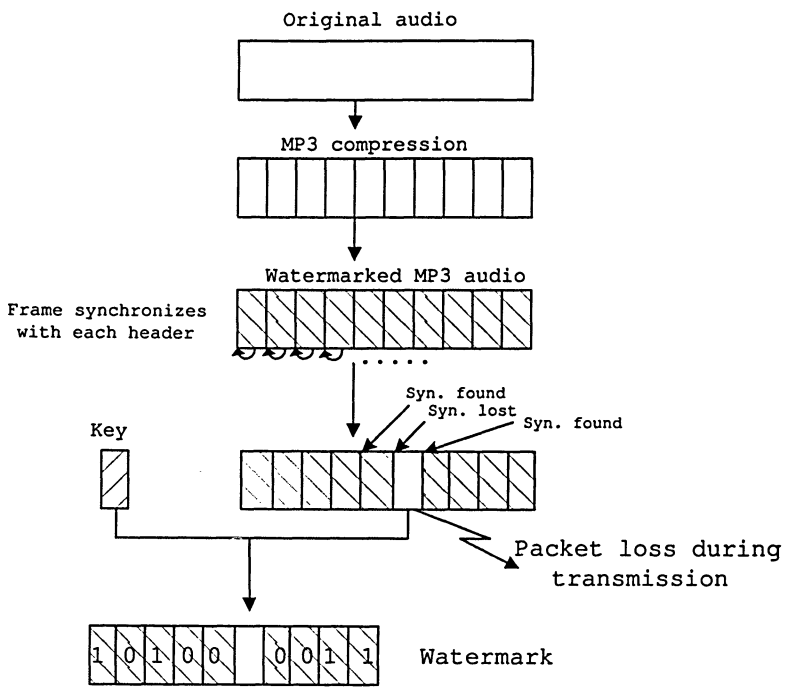


Figure 4.14: The concept of the proposed synchronization scheme. The watermark is extracted frame-by-frame. The key is the watermark spreading vector.

4.3.2 Watermark Extraction

When the system receives the watermarked signal $\tilde{y}$, it compares the distance between the two dither vectors:

$$D_1 = \sum_{i=1}^{L} (\tilde{y} - (q(x + d_i(1)) - d_i(1))),$$
$$D_2 = \sum_{i=1}^{L} (\tilde{y} - (q(x + d_i(2)) - d_i(2))). \tag{4.30}$$

Ideally D_1 is 0 or D_2 is 0 if there is no modification to the audio by any means. Thus if $D_1 < D_2$ the embedded watermark bit is 0; otherwise the embedded watermark bit is 1.

After the watermarking bits are retrieved from the MP3 audio, we need to define a scheme to verify that this is the correct watermark that was originally embedded into the audio. To address this problem, we add "pattern" bits into the system. At the head of

each watermarking payload, we add H-bit headers for synchronization purpose. Also we add one parity checking bit in every m bits in the watermarking payload. Then we insert n Position Identifiers (PI) evenly into the watermarking bits. Each PI uniquely identifies a sub-block of the watermark payload (Figure 4.15). These watermarking bit-streams, including the header and PIs, are further encoded by FEC coding scheme. The channel coded watermarking bits are embedded into the MP3 codec frames repeatedly until the end of the file. At the receiver side, we retrieve watermarks bit-stream out of Application Data Units (ADUs). If we experience packet loss in one sub-block, for example,the sub-block PI_i, first we attempt to recover the missing bits using FEC code. If the FEC code fails to recover all the lost bits, with the help of headers and PIs, we can use the same sub-block PI_i in next watermarking set to recover this sub-block. After we have decoded all the sub-blocks with all correct patterns, we get a full set of watermark successfully (Figure 4.16).

The coding efficiency of our scheme is:

$$e = \frac{P}{P + P/m + n * i + H + K}, \tag{4.31}$$

where P is the number of bits for watermarking payload, P/m is the number of bits for parity checking, n is the bit-length of PI, H is the length of headers, and K is the overhead of FEC code.

4.4 Applying Forward Error Correction (FEC) code

To compare and test different FEC codes, we applied two main types of classical coding schemes, Block coding (BCH, Reed-Solomon) and Convolutional coding. Furthermore, a new class of FEC code known as Turbo code is also implemented. The following is the description for each coding scheme.

4.4.1 BCH

Bose-Chaudhuri-Hocquenghem (BCH) is a kind of block coding scheme. A block coding scheme divide a bit stream into non-overlapping blocks and code each block independently.

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

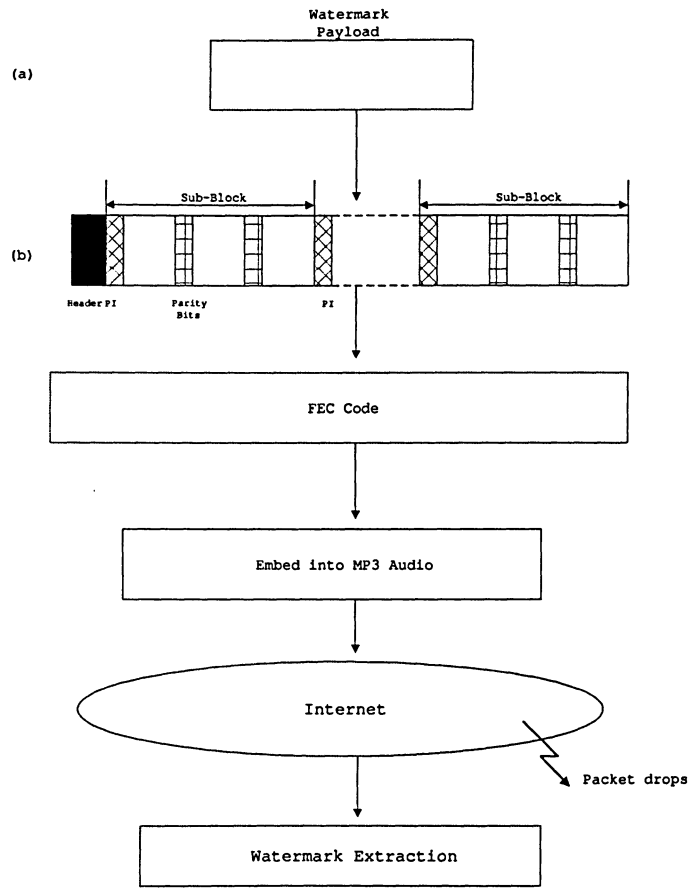


Figure 4.15: (a)Watermark Payload (b) Watermark Payload with Headers, PIs, and Parity Check bits

Block codes used in practical applications today belong to the class of linear cyclic codes, since these codes lend themselves to easier implementations. A coding scheme is referred to as being linear if the sum of two code vectors is also a code vector [66].

For any positive integers, $m \geq 3$ and $t < 2^{m-1}$, there is a binary BCH code with the following parameters:

- block length: $n = 2^m - 1$,
- number of parity check bits: $n - k \leq mt$,
- minimum distance: $d_{min} \geq 2t + 1$.

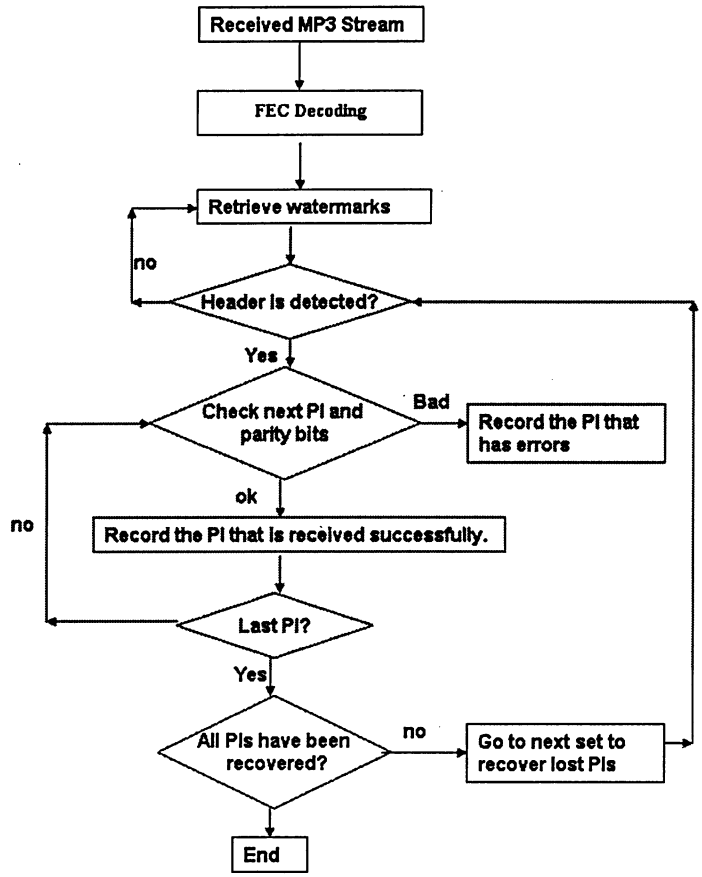


Figure 4.16: Recovery Flowchart

Each binary BCH code (n, k, t) can correct up to t-bit errors, and thus it is also referred to as a t-error-correcting code.

4.4.2 Reed-Solomon

Reed Solomon codes are a subset of BCH codes and are linear block codes as well. A Reed-Solomon code is specified as RS(n, k) with s-bit symbols [67].

If p is a primary number and q is any power of p, there exist BCH codes with q-ary symbols. For any choice of positive integer r and t, a q-ary BCH code is of length $n = q^r - 1$, which is capable of correcting any combination of t or fewer symbol errors and requires no

more than $2rt$ parity-check symbols. RS codes are a subclass of nonbinary BCH codes with $r = 1$. A (n, k, t) RS code with q-ary symbols has the following parameters:

- block length: $n = q - 1$,
- number of parity-check bits: $n - k = 2t$,
- minimum distance: $d_{min} = 2t + 1$.

For example: a popular Reed-Solomon code is RS(255,223) with 8-bit symbols. Each codeword contains 255 code word bytes, of which 223 bytes are data and 32 bytes are parity. For this code:

$n = 255$, $k = 223$, $s = 8$

$2t = 32$, $t = 16$

The decoder can correct any 16 symbol errors in the code word: i.e. errors in up to 16 bytes anywhere in the codeword can be automatically corrected.

BCH and RS coding schemes have a well defined algebraic structure, which has facilitated the development of efficient coding and decoding schemes. In addition, RS codes have optimal "distance properties", i.e., provide optimal error correction capabilities given a fixed number of parity bits, and excellent "burst error suppression" capabilities.

4.4.3 Convolutional

Large block sizes are impractical for channels with very slow data rates if the sender and receiver wish to communicate without substantial average delay. A code allows early decoding (decoding before an entire block is received) with good performance is needed.

Convolutional codes meet the requirement in that the encoder can be visualised as a linear sequential circuit (Figure 4.17). In a convolutional code, the characters are converted to a bitstream and then this bitstream is itself processed to add the error-reduction qualities. There is no relationship between the boundaries between characters and the error-reduction process. Since the channel errors are also not related in any way to the character boundaries,

convolutional codes are better suited to serial links than block codes, which were originally designed for protecting errors in memory banks and similar structures.

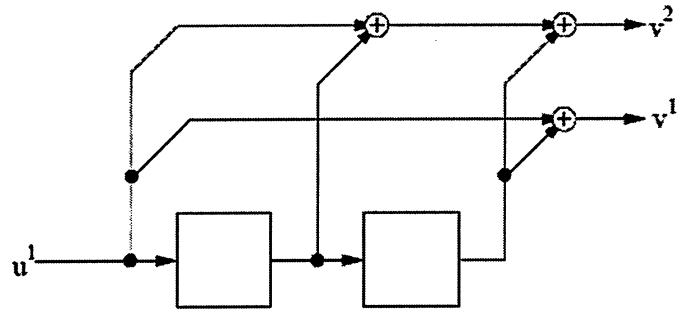


Figure 4.17: A diagram of a $R = 1/2$, $m = 2$ convolutional code in controller-canonical form. Each input bit leads to two output bits

Figure 4.17 shows a rate 1/2 binary convolutional encoder with $m = 2$. For each input bit, there are 2 output bits which depend on the previous 2 input bits. The encoder consists of an m-state shift register together with n modulo-2 adders and a multiplexer for serializing the encoder outputs. If the input sequence is $\mathbf{u} = (u_0, u_1, u_2, ...)$, the two encoder output sequence $\mathbf{v}^{(1)} = (\, v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, ...)$ and $\mathbf{v}^{(2)} = (\, v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, ...)$ are equal to the convolution of the input sequence $\mathbf{u}$ with the two code generator sequence $\mathbf{g}^{(1)}$ = (1,1,1) and $\mathbf{g}^{(2)}$ = (1,0,1), i.e., the encoding equations are $\mathbf{v}^{(1)} = u * g^{(1)}$ and $\mathbf{v}^{(2)} = u * g^{(2)}$.

4.4.4 Turbo Codes

The introduction of Turbo Codes at the International Conference on Communication (ICC) in 1993 brought the performance of practical coding closer to Shannon's theoretical specifications [68, 69, 7].

A Turbo encoder is a combination of two simple recursive convolutional encoders, each using a small number of states. For a block of k information bits, each constituent code generates a set of parity bits. The turbo code consists of the information bits and both sets of parity, as shown in Figure 4.18.

The key innovation is an interleaver P, which permutes the original k information bits

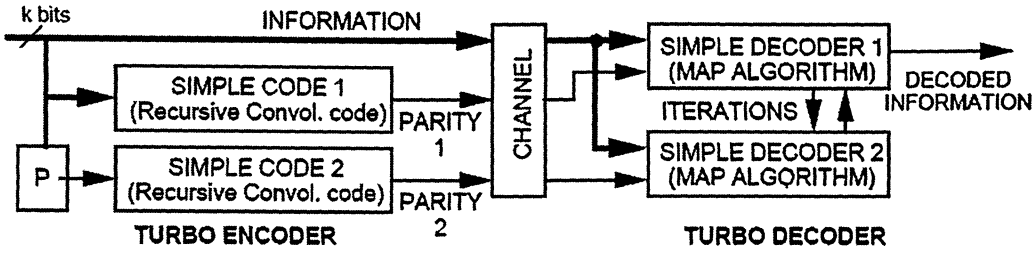


Figure 4.18: Diagram of Turbo encoder and decoder [7]

before encoding the second code. If the interleaver is well-chosen, information blocks that correspond to error-prone codewords in one code will correspond to error-resistant codewords in the other code. The resulting code achieves performance similar to that of Shannon's random codes. However, random codes approach optimum performance only at the price of a prohibitively complex decoder.

Turbo decoding uses two simple decoders individually matched to the simple constituent codes. Each decoders sends likelihood estimates of the decoded bits to the other decoder, and uses the corresponding estimates from the other decoder as a *priori* likelihoods. The constituent decoders use the "MAP" (maximum a *posteriori*) bitwise decoding algorithm, which requires the same number of states as the well-known Viterbi algorithm. The turbo decoder iterates between the outputs of the two decoders until reaching satisfactory convergence. The final output is a hard-quantized version of the likelihood estimate of either decoders.

Chapter 5

Experimental Result

The implementation of encoding and decoding of MP3 as well as the simulation of packet drop on network were all carried out in Matlab. The MP3 decoder can decode the stereo (dual-channel) MP3 audio with sampling rate of 44.1 kHz and bit rate of 32 - 320 kbit/s.

5.1 Results for Watermark Extraction

Our watermark payload including headers and position identifiers is 170 bits length, which are further FEC coded and embedded into the MP3 audio sequentially (Figure 4.15). In order to find the FEC code that best suits our need, we applied the popular coding system of BCH, Convolutional, Reed-Solomon and Turbo code respectively. Based on the Equations 4.12 and 4.27, we set the length of dither vector as 300 (See Table 5.1; the recovery result is the best when dither vector length L is 300 - 400), and quantize the lowest 300 coefficients (0 - 11000 Hz) in each MP3 granule and hide 1 bit every two granules, which is equivalent to 1 watermark bit per MP3 frame as each frame contains two granules. One MP3 frame covers 26 milli-seconds range of audio, so our embedding capacity is 38 bits/s. We could enhance the robustness and impercepbility by expanding the dither vectors across several MP3 frames (L is greater than 576). But the performance of watermark will become worse if the music is transmitted over a lossy network where packet loss happens. Thus we prefer to retain the extraction of watermark dependent on just one frame.

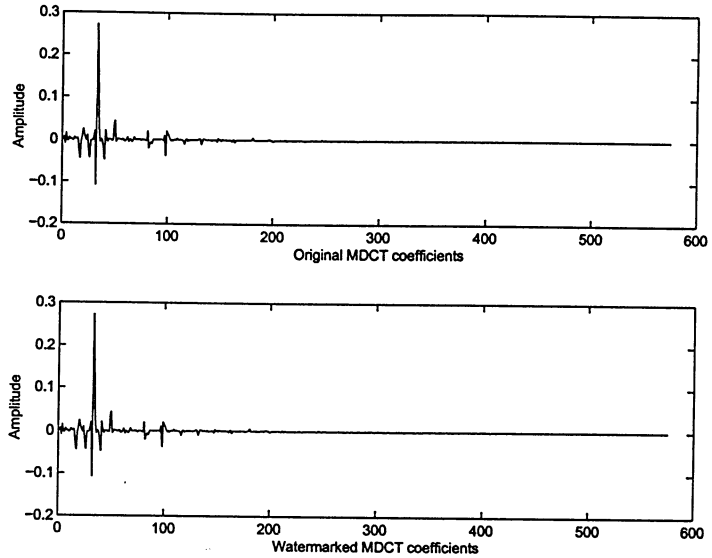


Figure 5.1: MDCT coefficients of un-watermarked and watermarked audio

5.1.1 Robustness and Attacks

To test the robustness of our watermarking scheme, we add Additive White Gaussian Noise (AWGN) to the host signal. Table 5.2 shows the recovery results, where in column 1 the host signal is not processed by any other means except adding noise; WNR is the Watermark-to-Noise Ratio; n means the watermark is unrecoverable with corresponding WNR.

Furthermore we use the third party audio processing tool Sony Sound Forge to manipulate

L-length	Recovery rate
50	53.77%
100	64.07%
150	76.88%
200	95.98%
250	97.99%
300	99.75%
350	99.50%
400	99.50%

Table 5.1: Selection of dither vector length L

WNR (dB)	Noise added directly to watermark[1]	Noise added in MDCT coefficients	Noise added in time domain
12.04	100%	98.49%	98.74%
9.03	100%	98.24%	98.74%
5.88	100%	95.48%	98.74%
3.01	100%	86.43%	98.74%
1.51	100%	72.86%	97.74%
0.60	100%	60.00%	97.74%
0.00	100%	55.00%	97.74%
-0.60	100%	n	97.74%
-1.51	89.42%	n	97.74%
-1.81	79.21%	n	97.74%
-3.013	52.00%	n	97.74%
-6.02	n	n	97.74%
-9.03	n	n	88.96%
-9.93	n	n	81.33%
-10.54	n	n	75.13%

Table 5.2: Recovery results after adding AWGN noise

the host signal whose results are shown in Table 5.3.

Based on the data in Table 5.2 and 5.3, we conclude that the proposed watermarking scheme is robust against normal signal manipulation and distortion. It performs better against time domain attacks than frequency domain (MDCT domain). Among all the signal processing methods, the echo addition and random cropping damage the watermark more than other operations. However they also noticeably degrade (or change) the sound quality. If they are used to attack the watermark, the host signal will be damaged as well.

5.2 Results After Packet Loss

We compared the bit error rate of watermarking that use ADU (Application Data Unit) and MP3 raw frames (Figure 5.3 and Figure 5.4). Theoretically the worst case of ADU will perform the same as MP3 raw frames because one physical packet lost actually renders one or more MP3 packets useless but causes just one ADU missing. If one ADU is lost, we will place the dummy frames to form MP3 stream. In our simulation scenarios, we randomly

	Music1		Music2		Music3	
	Normal	FEC	Normal	FEC	Normal	FEC
Normalize	98.99%	100%	85.43%	88.00%	84.81%	96.00%
DC shift	99.25%	100%	92.46%	100%	84.81%	96.00%
Echo (Delay 200ms/Decay 100ms)	76.37%	84.25%	67.05%	69.20%	69.21%	72.30%
Crop (Fade in 20ms/Fade out 20ms)	76.88%	82.50%	67.09%	68.00%	69.50%	75.21%
Invert	98.99%	100%	94.47%	100%	83.75%	95.00%
Chorus (Chorus out delay: 40ms)	90.45%	92.50%	82.41%	83.00%	82.80%	96.41%
Stereo-Mono Conversion	98.99%	100%	92.45%	100%	84.81%	96.00%
EQ	99.25%	100%	92.46%	100%	84.81%	94.00%
Insert Silence	85.58%	90.00%	80.38%	85.00%	71.34%	74.00%
Distortion	98.99%	100%	92.46%	100%	84.81%	96.00%
Smooth	98.99%	100%	92.46%	100%	84.81%	96.00%
MP3 (320 kb/s)	98.94%	100%	91.46%	100%	84.17%	96.00%
MP3 (256 kb/s)	99.25%	100%	91.96%	100%	84.00%	95.55%
MP3 (192 kb/s)	99.25%	100%	92.45%	100%	83.29%	95.50%
MP3 (128 kb/s)	99.25%	100%	92.46%	100%	83.29%	94.00%
MP3 (96 kb/s)	99.25%	100%	90.45%	100%	82.25%	98.00%
MP3 (64 kb/s)	97.20%	100%	90.20%	100%	82.00%	93.67%

Table 5.3: Recovery results after different signal processing (Music1 is a piano clip; music2 is a classical clip; music3 is a blues clip).

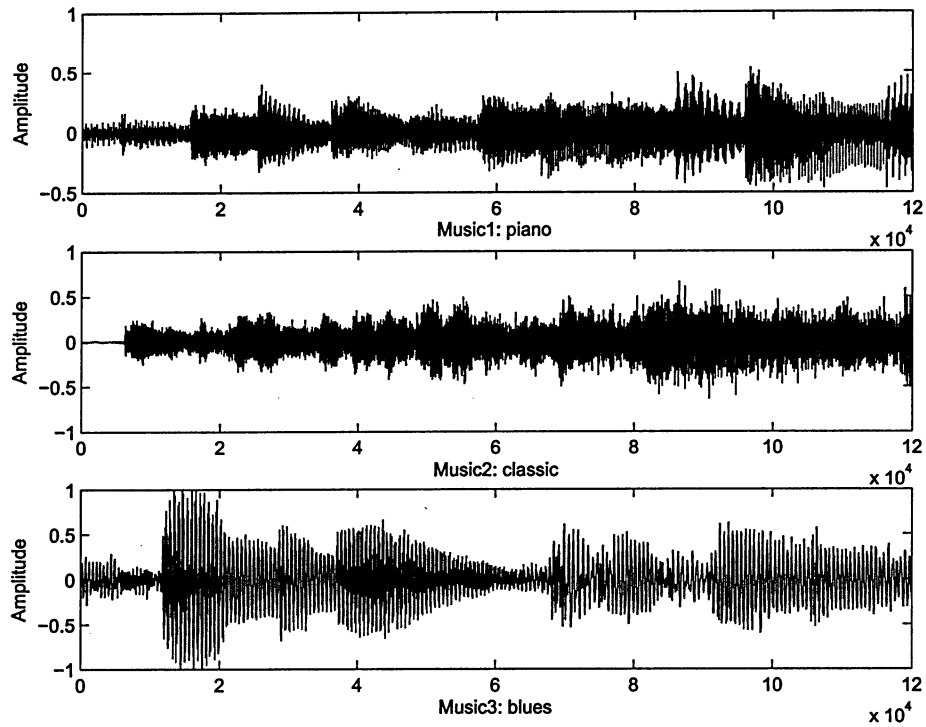


Figure 5.2: The audio samples that are used in the experiment

drop the packets with the drop probability of $\alpha = \{0.01, 0.02, 0.03, 0.04\}$. Table 5.4 shows that in our simulations ADU achieves almost half of the bit error rate of the raw MP3 frames (This result varies with different class of audios).

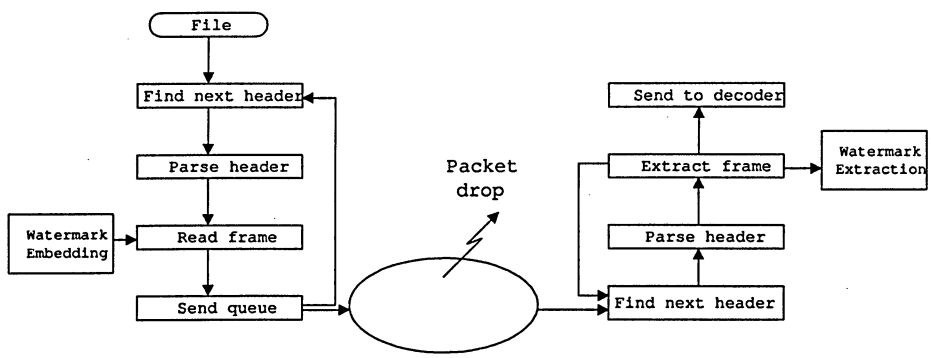


Figure 5.3: Packet drop without using ADU

In addition, we compare the audio quality when the MP3 audio experiences packet loss.

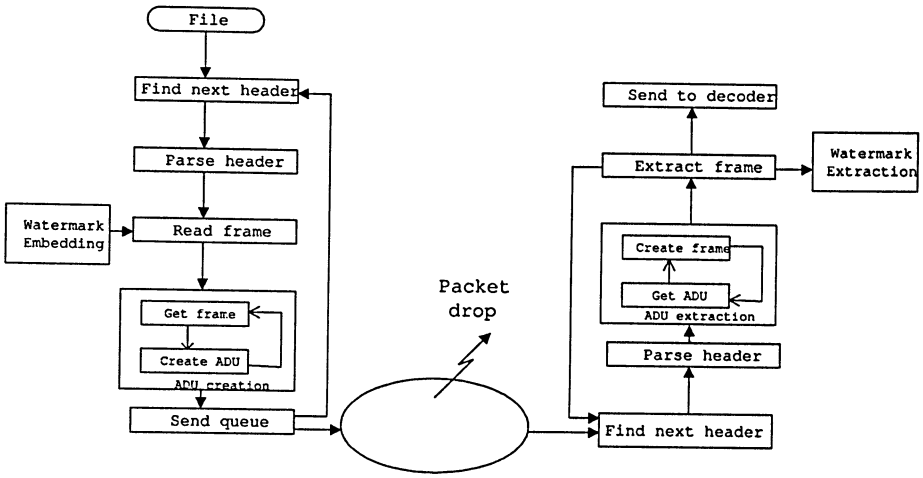


Figure 5.4: Packet drop using ADU

	Watermark bit-error-rate	
Packet drop rate	No ADU	ADU
0.01	0.019	0.011
0.02	0.032	0.019
0.03	0.055	0.025
0.04	0.056	0.035

Table 5.4: Comparison of watermark bit-error-rate before and after using ADU

Based on the experiment results, we find that the degradation of the audio quality is noticeable when the network starts to drop packets. The sound becomes annoying when the packet lost rate is 6% if raw MP3 frames are transmitted. Whereas if the ADU is applied the music retains almost the same quality when the packet loss rate is 10%.

Since the ADU outperforms the raw MP3 frames significantly in the lossy environment, we applied the ADU in following simulations.

These watermarked MP3 ADUs are then transmitted through the networks (simulated in Matlab). Again we drop the audio packets randomly in the middle.

To help us detect the packet loss occurred during transmission, we take advantage of the private bits in each MP3 frame header as defined in International Standard. The private bits are three-bit-length and can be customized by users. We insert sequence numbers as private bits in each frame before the transmission. The sequence number will be re-used every

eight frames. At the receiver side, the packet loss is detected when the sequence number is inconsistent. The lost detection will be failed if by chance exactly seven consecutive packets are missing, but this possibility is as low as 2.8×10^{-9} even when the packet loss rate is as high as 6%.

The missing watermarking bits in lost packet are refilled with zeros. Regarding the parameters used in the FEC coding system, we select the numbers n (block length) and k (number of message bits) that are most often used in common communication system (Table 5.5). Figure 5.5 is the recovery results of different schemes, from which it shows that even if there is no special FEC coding system applied, we still have the recovery rate as high as 96.5% when the packet drop rate α equals to 2%. With the packet drop rate becoming higher, Turbo code achieves the best recovery performance with the least overhead and best recovery capacity among all the systems. Even when the packet drop probability has reached 10%, Turbo Code still recovers more than 99% of the watermarking bits.

The above result is obtained as our expected, because as mentioned in Section 4.4.4, with multiple convolutional encoders and random permutation (interleaving) of input data, Turbo code can achieve performance similar to the Shannon's theoretical limit with random codes. However the above result does not consider the computational complexity of the FEC systems. If we take the computational time into account, the Convolutional Coding is the fastest scheme and the Turbo coding is the most time consuming scheme, which takes double the time of Reed-Solomon scheme when encoding the same amount of data. Therefore which coding system is the best depends on the application on hand. If the application can tolerate the delay, Turbo code is the best. Otherwise the Convolutional or BCH would be a better choice for packet recovery purposes.

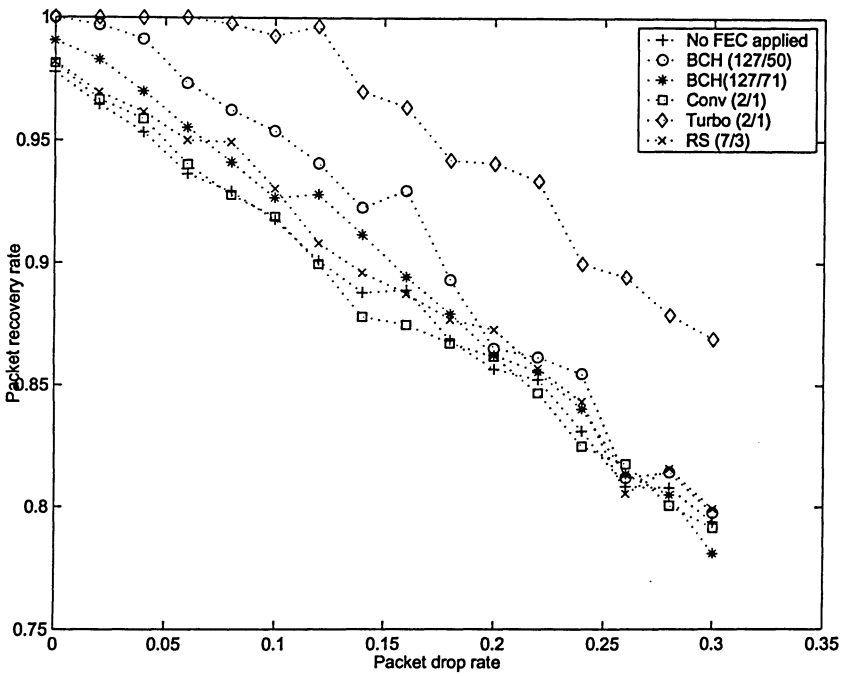


Figure 5.5: Watermark recovery with different packet drop rate

	NO FEC	BCH (127/50)	BCH(127/71)	Conv(2/1)	RS(31/13)	Turbo(2/1)
Overhead:	0	150%	79%	100%	138%	100%
Drop Prob. α						
0	0.9778	1.0000	0.9906	0.9814	1.000	1.0000
0.02	0.9645	0.9971	0.9829	0.9666	0.9984	1.0000
0.04	0.9533	0.9913	0.9700	0.9587	0.9946	1.0000
0.06	0.9366	0.9732	0.9552	0.9404	0.9869	1.0000
0.08	0.9294	0.9623	0.9412	0.9280	0.9897	0.9976
0.10	0.9178	0.9538	0.9269	0.9190	0.9795	0.9926
0.12	0.9009	0.9409	0.9283	0.8996	0.9771	0.9966
0.14	0.8881	0.9230	0.9116	0.8781	0.9509	0.9700
0.16	0.8890	0.9299	0.8944	0.8749	0.9328	0.9635
0.18	0.8686	0.8933	0.8794	0.8675	0.9357	0.9423
0.20	0.8568	0.8653	0.8626	0.8619	0.9235	0.9410
0.22	0.8525	0.8617	0.8556	0.8469	0.9042	0.9339
0.24	0.8311	0.8548	0.8402	0.8250	0.8690	0.8998
0.26	0.8086	0.8121	0.8135	0.8177	0.8905	0.8944
0.28	0.8080	0.8144	0.8052	0.8008	0.8628	0.8790
0.30	0.7941	0.7978	0.7811	0.7916	0.8373	0.8692

Table 5.5: Watermark detection rate in lossy environment

FEC Coding Scheme	Overhead	Relative CPU Time
BCH(127/50)	154%	2.855
BCH(127/71)	78.9%	1.87
Conv(2/1)	100%	0.15
RS(7/3)	133%	168.78
Turbo(2/1)	100%	271.85

Table 5.6: Comparison of Overhead and CPU Time. The CPU time reflects the time that is used by the MATLAB process when conducting each individual operation. The results are obtained on a Pentium III 800 MHz platform.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

Internet multimedia applications are becoming more and more popular especially for MP3 audios. In this thesis, we have proposed a novel and robust scheme that embeds and recovers digital audio watermarking for MP3 stream through Computer Network.

We implement MP3 audio decoder based on ISO/IEC 11172-3 standard. The decoder covers the whole decoding process including Huffman decoding, requantization, IMDCT, and synthesis polyphase filterbank.

The watermark is embedded in MDCT coefficients of MP3 audio using Quantization Index Modulation (QIM) scheme. QIM is a kind of quantization algorithm that has been used in image or video watermarking which provides higher information-embedding rate than previously proposed spread-spectrum and low-bit modulation systems. To the best of our knowledge, no research has been done to apply the QIM to MP3 compressed domain. The experimental results indicate that our proposed scheme is robust under normal signal distortion and the host audio quality is not degraded significantly.

We have simulated the packet drops to the watermarked MP3 audio and recovered the lost watermark caused by the packet loss. The Application Data Unit (ADU) is also introduced to the system that has been proven to reduce the negative impact of the packet drop during transmission.

FEC code is further applied in our recovery scheme to improve the performance. Four

types of FEC codes are implemented among which Turbo code offered the best recovery capacity in the simulation experiment.

This scheme is very advantageous to transmit watermarks in lossy environment. Under normal networking conditions ($\alpha < 6\%$), we can successfully recover full set of watermarks once we have received enough amount of MP3 frames. Even when the packet loss is as high as 10%, we still have more than 99% watermarks survived.

6.2 Future Works

(1). Variable Bit Rate of MP3

The MP3 standard specifies two different types of bitrates; Constant Bitrate (CBR) and Variable Bitrate (VBR). VBR allows the bitrate to vary depending on the dynamics of the signal. The quality is set using a threshold specified by the user to inform the encoder of the maximum bitrate allowed. For example, near the beginning and ending of a song (assuming it starts and ends softly), where the volume is lower, and the music is less "demanding" in terms of its encodability, VBR drops the bit rate, simply because there's not much there to encode, and the wasted space is not necessary. In the middle of the song, where it may be more complicated, VBR increases the bitrate to retain the enriched melody. It may end up with a file that's the same overall size as a 160 kbit/s CBR, but that uses frames as low as 32 on the really dead parts, and as high as 320 on the really tough parts. The bitrate is dynamically adapting to keep the quality constant. Unfortunately there are some drawbacks of using VBR. Firstly, VBR might cause timing difficulties for some decoders, i.e. the MP3 player might display incorrect timing information or non at all. Secondly, CBR is often required for broadcasting, which initially was an important purpose of the MP3 format [4].

VBR format is not widely used like CBR but recently more and more decoders start to support VBR format because of its advantage. VBR will also be a better format when delivering streaming MP3 through networks. So migrating our proposed scheme to VBR MP3 will improve the overall performance of our scheme.

(2). Watermark and Rights Expression Language (REL)

The new standard of MPEG-21 specifies a machine-readable language called REL that can declare rights and permissions using the terms defined in the Rights Data Dictionary. The REL provides a very flexible mechanism to ensure the data is processed in accordance with individual rights. It defines the language's syntax and semantics, but it does not describe specifications for security in trusted systems, propose specific applications, or provide the details of the accounting systems required.

Indeed we can combine the REL with our watermarking scheme, define the watermarking message, extract and check it based on REL. It would be very beneficial to improve the standardization, reliability and interoperability of the watermarking schemes.

Appendix A

List of Acronyms

AAC	Advanced Audio Coding
ADU	Application Data Unit
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
CBR	Constant Bit rate
DRM	Digital Rights Management
FEC	Forward Error Correction
FFT	Fast Fourier Transform
IMDCT	Inverse Modified Discrete Cosine Transform
ISO	International Standardization Organization
MDCT	Modified Discrete Cosine Transform
MP3	MPEG-1 Layer III
MPEG	Motion Pictures Expert Group
PCM	Pulse Code Modulation
QIM	Quantization Index Modulation
RDD	Rights Data Dictionary
REL	Rights Expression Language
RS	Reed-Solomon
SCS	Scalar Costa Scheme
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
SS	Spread Spectrum
STDM	Spread Transform Dither Modulation
VBR	Variable Bit Rate
WNR	Watermark-to-Noise Ratio

Bibliography

[1] P. Noll. MPEG digital audio coding. *IEEE Signal Processing Magazine*, pages 59–81, September 1997.

[2] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of IEEE*, 88(4):451–513, August 1993.

[3] H.C. Lai. Real-time implementation of MPEG-1 layer 3 audio decoder on a DSP chip. *Master thesis, College of Electrical Engineering and Computer Science, National Chiao-Tung University, Taiwan*, 2001.

[4] R. Raissi. The theory behind mp3. http://mpgedit.org/mpgedit/docs.html.

[5] K. Lagerstrom. Design and implementation of an MPEG-1 layer III audio decoder. *Master Thesis. Computer Science and Engineering Program, Chalmers University of Technology, Gothenburg, Sweden*, 2001.

[6] B. Chen and G.W.Wornell. Quantization Index Modulation Methods: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans, Info. Theory*, 47(4):1423–1443, May 2001.

[7] S. Dolinar, D. Divsalar, and F. Pollara. Turbo codes and space communications. *Communications Systems and Research Section, Jet Propulsion Laboratory, California Institute of Technology.*

[8] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(4):1062–1078, July 1999.

[9] S. Esmaili, S. Krishnan, and K. Raahemifar. Audio watermarking based on time-frequency characteristics. *IEEE Canadian Journal on Electrical Comput. Eng*, 28:57–61, July 2003.

[10] C.I. Podilchuk and E.J. Delp. Digital watermarking: algorithms and applications. *Signal Processing Magazine, IEEE*, 18:33 – 46, July 2001.

[11] L. Boney, A. Tewfik, and K. Hamdy. Digital watermarks for audio signals. *EUSIPCO-96, VIII European Signal Proc. Conf.*, pages 473–480, September 1996.

[12] N. Cvejic and T. Seppanen. Audio watermarking using attack characterisation. *Electronics Letters, 26th*, 39(13):1020– 1021, June 2003.

[13] J. Seok and J. Hong. Audio watermarking for copyright protection of digital audio data. *Electronics Letters, 4th*, 37(1):60–61, January 2001.

[14] ISO/IEC 11172-3. Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s - part 3.

[15] M. Seadle, J.R. Deller Jr, and A. Gurijala. Why watermark? The copyright need for an engineering solution. *International Conference on Digital Libraries. Proceedings of the 2nd ACM/IEEE-CS joint conference*, pages 324 – 325, 2002.

[16] ISO/IEC JTC1/SC29/WG11. MPEG-2. http://www.chiariglione.org/mpeg/standards /mpeg-2 /mpeg-2.htm.

[17] ISO/IEC JTC1/SC29/WG11 N4668. Overview of the MPEG-4 standard. http:// www.chiariglione.org /mpeg/standards/mpeg-4/mpeg-4.htm.

[18] ISO/IEC JTC1/SC29/WG11/N5231. MPEG-21 overview v.5 http://www. chiariglione.org/ mpeg/standards/mpeg-21/mpeg-21.htm.

[19] J. Bormans, J. Gelissen, and A. Perkis. MPEG-21: The 21st century multimedia framework. *Signal Processing Magazine, IEEE*, 20:53–62, March 2003.

[20] J. Delgado, I. Gallego, and E. Rodriguez. Use of the MPEG-21 rights expression language for music distribution. *Web Delivering of Music, 2003. 2003 WEDELMUSIC. Proceedings. Third International Conference on*, pages 15–17, September 2003.

[21] S. Shlien. Guide to MPEG-1 audio standard. *IEEE transactions on Broadcasting*, 40(4):206–218, December 1994.

[22] D. Pan. A tutorial on MPEG/Audio compression. *IEEE Multimedia Journal*, 2:60–74, 1995.

[23] K. Brandenburg. MP3 and AAC explained. *AES 17th International Conference on High Quality Audio Coding*, 1999.

[24] R. Finlayson. A more loss-tolerant RTP payload format for MP3 audio. *IETF RFC3119*, 2001.

[25] P. Arttameeyanant, P. Kumhom, and K. Chamnongthai. Audio watermarking for Internet. *Industrial Technology, 2002. IEEE ICIT '02.*, 2:976 – 979, December 2002.

[26] I.K. Yeo and H.J. Kim. Modified patchwork algorithm: a novel audio watermarking scheme. *IEEE Transactions, Speech and Audio Processing*, 11:381 – 386, July 2003.

[27] X. Li, M. Zhang, and R. Zhang. A new adaptive audio watermarking algorithm. *Fifth World Congress, Intelligent Control and Automation, 2004. WCICA 2004.*, 5:4357 – 4361, June 2004.

[28] N. Cvejic, D. Tujkovic, and T. Seppanen. Increasing robustness of an audio watermark using turbo codes. *IEEE International Conference on, Multimedia and Expo, 2003. ICME '03. Proceedings.*, 1:217–220, July 2003.

[29] W.N. Lie and L.C. Chang. Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking. *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, 2:45–48, May 2001.

[30] D.K Koukopoulos and Y.C Stamatiou. A compressed-domain watermarking algorithm for MPEG audio layer 3. *ACM Multimedia 2001 Workshops-Multimedia and Security*, pages 7–10, 2001.

[31] J.M. Boyce and R.D. Gaglianello. Packet loss effects on MPEG video sent over the public Internet. *Proceedings of the sixth ACM international conference on Multimedia*, pages 181–190, 1998.

[32] R. Parviainen. Error resilience methods for multicast MPEG-1 audio transmission. *Master Thesis. Department of computer science and electrical engineering, Lulea University of Technology, Sweden*, 2001.

[33] C. Perkins, O. Hodson, and V. Hardman. A survey of packet loss recovery techniques for streaming audio. *Network, IEEE*, 12:40–48, September 1998.

[34] B.W. Wah, X. Su, and D. Lin. A survey of error-concealment schemes for real-time audio and video transmission over Internet. *Proceedings IEEE International Symposium on Multimedia Software Engineering*, page 17, December 2000.

[35] N. Feamster and H. Balakrishnan. Packet loss recovery for streaming video. *12th International Packet Video Workshop (PV2002), Pittsburgh, PA*, April 2002.

[36] J. G. Apostolopoulos. Watercasting: Distributed watermarking of multicast media. *Proceedings SPIE Visual Communications and Image Processing*, 4310:392–409, January 2001.

[37] E.T. Lin, C.I. Podilchuk, T. Kalker, and E.J. Delp. Streaming video and rate scalable compression: what are the challenges for watermarking? *Journal of Electronic Imaging, SPIE 2004*, 13:198–208, January 2004.

[38] C.Y. Lin, D. Sow, and S.F. Chang. Using self-authentication and recovery images for error concealment in wireless environments. *Proceedings of SPIE, Multimedia Systems and Applications IV*, 4518:267–274, August 2001.

[39] I. Brown, C. Perkins, and J. Crowcroft. Watercasting: Distributed watermarking of multicast media. *Networked Group Communication: First International COST264 Workshop, NGC'99*, November 1999.

[40] H. J. Kim. Audio watermarking techniques. *Pacific Rim Workshop on Digital Steganography, Kyushu Institute of Technology, Kitakyushu, Japan*, July 2003.

[41] I.J. Cox, J. Kilian, F.T. Leigton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing*, 6:1673–1687, 1996.

[42] N. Cvejic, A. Keskinarkaus, and T. Seppanen. Audio watermarking using m-sequences and temporal masking. *IEEE workshops on Applications of Signal Processing to Audio and Acoustics, New Paltz, New York*, 6:227–230, 2001.

[43] D. Kirovski and H.S. Malvar. Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing, special issue on data hiding*, 51:1020 – 1034, April 2003.

[44] I.K. Yeo and H.J. Kim. Modified Patchwork Algorithm: A novel audio watermarking scheme. *IEEE Transactions on Speech and Audio Processing*, 11:381 – 386, July 2003.

[45] W. Bender, D. Gruhl, Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 38:313–336, 1996.

[46] D. Gruhl, W. Bender, and A. Lu. Echo hiding. *Pre-proceedings: Information hiding, Cambridge, UK*, pages 295 – 316, 1996.

[47] O.O. Hyen, W.S. Jong, W.H. Jin, and H.Y. Dae. New echo embedding technique for robust and imperceptible audiowatermarking. *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01), IEEE*, 3:1341 – 1344, 2001.

[48] H.J. Kim and Y.H. Choi. A novel echo hiding algorithm. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003.

[49] M.F. Mansour and A.H. Tewfik. Audio watermarking by time-scale modification. *International Conference on Acoustics, Speech, and Signal Processing*, 3:1353 – 1356, 2001.

[50] J. Foote, J. Adcock, and A. Girgensohn. Time base modulation: A new approach to watermarking audio. *IEEE International Conference on Multimedia and Expo*, 1:221 – 224, July 2003.

[51] C.P. Wu, P.C. Su, and C.J. Kup. Robust and efficient digital audio watermarking using audio content analysis. *Security and Watermarking of Multimedia Contents*, 3971:382–392, 2000.

[52] M.F. Mansour and A.H. Tewfik. Time-scale invariant audio data embedding. *International Conference on Multimedia and Expo*, 2001.

[53] B. Chen and G.W.Wornell. Dither Modulation: A new approach to digital watermarking and information embedding. *Proceedings of SPIE: Security and Watermarking of Multimedia Content, San Jose, CA*, 3657:342–353, January 1999.

[54] B. Chen and G.W.Wornell. Quantization Index Modulation methods for digital watermarking and information embedding of multimedia. *Journal of VLSI Signal Processing, 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.*, 27:7–33, 2001.

[55] J.J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod. Scalar costa scheme for information embedding. *IEEE Trans. On Signal Processing*, 51:1003–1019, April 2003.

[56] J.J. Eggers, J. Su, and B. Girod. A blind watermarking scheme based on structured codebooks. *Secure Images and Image Authentication (Ref. No. 2000/039), IEEE Seminar*, pages 4/1–4/21, 2000.

[57] M. Yajnik, J. Kurose, and D. Towsley. Packet loss correlation in the mbone multicast network. *Global Telecommunications Conference, 1996. GLOBECOM '96*, pages 94–99, November 1996.

[58] C. Perkins, O. Hodson, and V. Hardman. A survey of packet-loss recovery techniques for streaming audio. *IEEE Network*, 12:40–48, September 1998.

[59] D.E. Comer. Internetworking with TCP/IP, Volume i: Principles, Protocols and Architecture. *Prentice Hall*, 1995.

[60] W. Li, X. Xue, X. Li, and P. Lu. A novel feature-based robust audio watermarking for copyright protection. *Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC*, pages 554–558, April 2003.

[61] C. Hsieh and P.Y. Sou. Blind cepstrum domain audio watermarking based on time energy features. *Digital Signal Processing, DSP 2002. 14th International Conference*, 2:705–708, July.

[62] W. Li, X.Y. Xue, and P.Z Lu. Robust audio watermarking based on rhythm region detection. *Electronics Letters*, 41:218–219, February 2005.

[63] S.W. Foo, F. Xue, and M. Li. A blind audio watermarking scheme using peak point extraction. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium*, 5:4409–4412, May 2005.

[64] J. Huang, Y. Wang, and Y.Q. Shi. A blind audio watermarking algorithm with self-synchronization. *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium*, 3:627–630, May 2002.

[65] D. Kirovski and H. Malvar. Robust spread-spectrum audio watermarking. *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01), IEEE International Conference*, 3:1345–1348, May 2001.

[66] H. Liu, H. Ma, M.E. Zaiki, and S. Gupta. Error correction schemes for networks: an overview. *Mobile Networks and Applications 2*, pages 167–182, 1997.

[67] http://www.4i2i.com/reed_solomon_codes.htm, An introduction to Reed-Solomon codes: principles, architecture and implementation.

[68] C. Berrou, A. Glavieux, and P Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. *IEEE Proceedings of the Int. Conf. on Communications, Geneva, Switzerland*, pages 1064–1070, May 1993.

[69] A. Jemibewon. A smart implementation of Turbo decoding for improved power efficiency. *Master Thesis, Virginia Polytechnic Institute and State University*, 2000.