

Efficient Watermarking Schemes for Speaker Verification Guaranteeing Non-repudiation

Thesis submitted to
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
in partial fulfillment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY
under the Faculty of Technology by

Remya A R
Register No: 3880

Under the Guidance of
Dr. A Sreekumar



Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India

June 2014

**Efficient Watermarking Schemes for Speaker Verification
Guaranteeing Non-repudiation**

Ph.D. thesis in the field of Audio Watermarking

Author:

Remya A R,
Research Scholar (Inspire Fellow),
Department of Computer Applications,
Cochin University of Science and Technology,
Kochi - 682 022, Kerala, India,
Email: remyacusat@gmail.com

Supervisor:

Dr. A. Sreekumar,
Associate Professor,
Department of Computer Applications,
Cochin University of Science and Technology,
Kochi - 682 022, Kerala, India.
Email: askcusat@gmail.com

***Cochin University of Science and Technology,
Kochi - 682 022, Kerala, India.
www.cusat.ac.in***

June 2014

To Achan & Amma

Dr. A. Sreekumar
Associate Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

04th June 2014

Certificate

Certified that the work presented in this thesis entitled “*Efficient Watermarking Schemes for Speaker Verification Guaranteeing Non-repudiation*” is based on the authentic record of research carried out by *Ms. Remya A R* under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi-682 022 and has not been included in any other thesis submitted for the award of any degree.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2556057, +91 484 2862395 Email: askcusat@gmail.com

Dr. A. Sreekumar
Associate Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

04th June 2014

Certificate

Certified that the work presented in this thesis entitled “*Efficient Watermarking Schemes for Speaker Verification Guaranteeing Non-repudiation*” submitted to Cochin University of Science and Technology by *Ms. Remya A R* for the award of degree of Doctor of Philosophy under the faculty of Technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2556057, +91 484 2862395 Email: askcusat@gmail.com

Declaration

I hereby declare that the work presented in this thesis entitled “*Efficient Watermarking Schemes for Speaker Verification Guaranteeing Non-repudiation*” is based on the original research work carried out by me under the supervision and guidance of *Dr. A. Sreekumar*, Associate Professor, Department of Computer Applications, Cochin University of Science and Technology, Kochi-682 022 and has not been included in any other thesis submitted previously for the award of any degree.

Remya A R

Kochi- 682 022
04th June 2014

Acknowledgments

First and foremost I bow in reverence before the Lord Almighty for helping me complete this endeavor.

This thesis is the fulfillment of my desire of research that formed in early studies of postgraduate Software Engineering. In the longest path of the desire, I am indebted to lord almighty and lot of individuals around me for the realization of the thesis and without whom it would not have been possible.

I would like to express my overwhelming gratitude to my research supervisor Dr. A.Sreekumar, Associate Professor, for his expertise shown in guiding my work and the willingness to share his knowledge and experience. He has given immense freedom for me in developing ideas and he is always willing to hear and acknowledge sincere efforts. His unsurpassed knowledge and critical but valuable remarks led me to do a good research. I would like to express my sincere gratitude to him for his prompt reading and careful critique of my thesis.

I would like to thank Department of Science and Technology (DST), Government of India for the INSPIRE fellowship program which funded my internship at the Department of Computer Applications, CUSAT.

Besides my supervisor, I would like to thank Dr B.Kannan (Associate Professor, Head of the Department), Dr. K.V. Pramod (Associate Professor), Dr.M.Jathavedan (Emeritus Professor) and Ms. Malathy S (Assistant Professor) of Department of Computer Applications, CUSAT for their encouragement, insightful comments and hard questions.

I remember with gratitude Dr. Supriya M.H from the Department of Electronics, CUSAT who helped me to comprehend the field of Signal Processing. I also thank Dr. Thomaskutty Mathew (School of Technology and Applied Sciences, Mahatma Gandhi University Regional Center), Dr. Madhu S.Nair (Department of Computer Science, University of Kerala) and Dr. Tony Thomas (Indian Institute of Information Technology and Management, Kerala) who facilitated me to better value the research area of Audio Watermarking.

It was a pleasure and wonderful experience working with my research team and I am grateful to all my fellow researchers especially Binu V.P, Cini Kurian, Jessy George, Jomy John, Santhosh Kumar M B, Simily Joseph, Sindhumol S, Sunil Kumar R and Tibin Thomas, for their queries which helped me to improve and in some case amend the proposed work. I would like to thank Dr.Dann V.J (Department of Physics, Maharajas College, Ernakulam), Dr.G.N.Prasanth (Department of Mathematics, Govt. College, Chittur, Palakkad), Mr. Ramkumar R & Mr. Bino Sebastian for their help in reviewing and formatting the thesis.

I want to specially mention my brother, for his endless support that helps me to reach my destination and my father in-law & sister in-law for their tremendous effort in reviewing and finalizing my thesis. I specially thank Mr. Praveesh K.M., Senior Industrial Designer at Axiom Consulting, Bangalore, for designing the cover page of the thesis.

The Department of Computer Applications and library facilities in CUSAT provided me the computing resources and supportive environment for this project. I am thankful to the academic, non-academic and technical staff for their indispensable support. I would like to thank the Open Source programming community for providing the appropriate tools for the creation of this thesis.

Finally, I thank my parents and my in-laws for their constant support and encouragements. Your prayer for me was what sustained me this far. My utmost gratitude to own family: Mithun, Eeshwar and Eesha for taking my actions optimistic and always supporting me.

*I dedicate this thesis
to
My Family
for their constant support and unconditional love.
I love you all dearly.*

Remya A R

Contents

Abstract	xxix
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Objectives of the Proposed Study	3
1.4 Scope of the Work	4
1.5 System Framework	5
1.6 Thesis Contributions	7
1.6.1 List of Research Papers	9
1.7 Thesis Outline	10
1.8 Summary	13
2 An Overview of Signal Processing & Watermarking	15
2.1 Introduction	15
2.2 Signals and Systems	15
2.3 Audio Signals	17
2.3.1 Description	17
2.3.2 Characteristics	18

2.3.3	Representation	20
2.3.4	Features	21
2.4	Overview of Human Auditory System	31
2.4.1	Frequency Masking	32
2.4.2	Temporal Masking	33
2.5	Speech and Audio Signal Processing	34
2.6	Frequency Component Analysis of Signals	35
2.6.1	The Fourier Transform	35
2.6.2	Hadamard Transform	39
2.7	Watermarking	41
2.7.1	General Model of Digital Watermarking	41
2.7.2	Statistical Model of Digital Watermarking	42
2.7.3	Communication Model of Digital Watermarking	43
2.7.4	Geometric Model of Digital Watermarking	44
2.8	Evaluating Watermarking Systems	45
2.8.1	The Notion of “Best”	46
2.8.2	Benchmarking	46
2.8.3	Scope of Testing	47
2.9	Summary	48
3	Review of Literature	49
3.1	Introduction	49
3.2	Review of Audio Watermarking Algorithms	49
3.2.1	Time-Domain Based Algorithms	50
3.2.2	Transformation Based Algorithms	56
3.2.3	Hybrid Algorithms	64
3.3	Evaluation Strategy of Watermarking Schemes	76
3.3.1	A Quantitative Approach to the Performance Evaluation	77

3.4	Summary	85
4	Data Collection and Feature Extraction	87
4.1	Introduction	87
4.2	Data Collection	88
4.3	Pre-Processing	88
4.4	Feature Extraction	98
4.4.1	Mel-Frequency Cepstral Coefficients (MFCC)	100
4.4.2	Spectral Flux	103
4.4.3	Spectral Roll-Off	104
4.4.4	Spectral Centroid	105
4.4.5	Energy entropy	106
4.4.6	Short-Time Energy	107
4.4.7	Zero-Cross Rate	109
4.4.8	Fundamental Frequency	110
4.5	Summary	110
5	Speaker Recognition - Verification and Identification	113
5.1	Introduction	113
5.2	Speaker Recognition	114
5.3	A Brief Review of Literature	117
5.4	Verification Process	121
5.5	Speaker Recognition with Artificial Neural network	123
5.5.1	Training Data	124
5.5.2	Testing Data	126
5.5.3	Experimental Results	127
5.6	Speaker Recognition with k-Nearest Neighbor and Support Vector Machine Classifiers	130
5.6.1	k-NN Classifier	130

5.6.2	SVM Classifier	130
5.6.3	Training Data	131
5.6.4	Testing Data	132
5.6.5	Experimental Results	133
5.7	Comparative Study of ANN,k-NN and SVM	138
5.8	Summary	140
6	Barcode Based FeatureMarking Scheme	143
6.1	Introduction	143
6.2	Fourier Analysis	144
6.3	One Dimensional and Two Dimensional Data Codes . . .	150
6.4	Proposed Scheme	150
6.4.1	Watermark Preparation	151
6.4.2	FeatureMark Embedding	151
6.4.3	Repeat Embedding	155
6.4.4	Signal Reconstruction	156
6.4.5	Watermark Detection	158
6.4.6	Digital Watermark Extraction	158
6.5	Experimental Results	161
6.5.1	Non-Repudiation Services	169
6.6	Summary	170
7	Data Matrix Based FeatureMarking Scheme	171
7.1	Introduction	171
7.2	Data Matrix Code - A type of Data Code	172
7.3	Proposed Scheme	172
7.3.1	Watermark Preparation	173
7.3.2	Synchronization Code Generation	174
7.3.3	Embedding Method	175

7.3.4	Repeat Embedding	181
7.3.5	Signal Reconstruction	181
7.3.6	FeatureMark Detection Scheme	181
7.3.7	Digital Watermark Extraction	184
7.4	Experimental Results	186
7.5	Summary	194
8	FeatureMarking with QR Code	195
8.1	Introduction	195
8.2	Quick Response (QR) Code	195
8.3	Walsh Analysis	196
8.4	Proposed Scheme	199
8.4.1	Watermark Preparation	200
8.4.2	Synchronization Code Generation	201
8.4.3	Embedding Method	202
8.4.4	Repeat Embedding	208
8.4.5	Signal Reconstruction	208
8.4.6	FeatureMark Detection Scheme	209
8.4.7	Digital Watermark Extraction	212
8.5	An Enhanced FeatureMarking Method	214
8.6	Experimental Results	215
8.6.1	The Goal: Guaranteeing Non-repudiation services	223
8.6.2	Merits and Demerits of the Proposed Schemes	223
8.7	Summary	224
9	Conclusions and Future Works	225
9.1	Brief Summary	225
9.2	Comparison with Existing Schemes	226
9.3	Contributions	228

9.4	Future Scope	229
9.5	Summary	229
A	Notations and abbreviations used in the thesis	231
B	List of Publications	237

List of Figures

1.1	Watermark embedding scheme	6
1.2	Watermark detecting scheme	7
2.1	Digital signal processing	17
4.1	Speech signal - Waveform representation	90
4.2	Speech signal - Spectrum	90
4.3	Speech signal - Spectrogram	91
4.4	Single frame	91
4.5	Single window	92
4.6	Hamming window	93
4.7	Rectangular window	94
4.8	Comparison between rectangular and hamming windows . .	95
4.9	Representation of a speech signal, its frames and the feature vectors	97
4.10	Feature extraction	99
4.11	MFCC feature extraction	100
4.12	Mel-cepstrum in time domain	102
4.13	MFCC graph	102
4.14	Spectral flux feature graph	103

4.15	Spectral roll-off feature graph	105
4.16	Centroid value plot	106
4.17	Entropy feature plot	107
4.18	Energy plot for a single frame	108
4.19	Plot of ZCR values	109
5.1	Male voiced speech	115
5.2	Female voiced speech	116
5.3	Speaker verification process	122
5.4	Representation of MSE and %E	129
5.5	Performance plot of ANN	129
5.6	k-NN speaker verification	134
5.7	SVM speaker verification	135
5.8	k-NN speaker identification	136
5.9	SVM speaker identification	136
5.10	k-NN speaker recognition	137
5.11	SVM speaker recognition	138
6.1	Amplitude-time plot	145
6.2	Spectrum of the signal	145
6.3	Amplitude and frequency plots	146
6.4	Intensity-time plot	148
6.5	Spectrogram	148
6.6	Sample barcode	151
6.7	Arnold transformed barcode	152
6.8	Watermark embedding scheme	155
6.9	Watermark extraction scheme	159
6.10	Sample 1	161
6.11	Sample 2	162

6.12	Single channel - original and FeatureMarked speech signal .	162
6.13	Multi-channel - original and FeatureMarked speech signal .	162
6.14	Average recovery rate	168
7.1	Sample data matrix code	173
7.2	Construction of embedding information	174
7.3	Barker codes	175
7.4	Arnold transformed data matrix code	177
7.5	Data matrix embedding scheme	178
7.6	Synchronization code detection	182
7.7	Watermark extraction scheme	184
7.8	Sample 1	187
7.9	Sample 2	187
7.10	Sample 3	187
7.11	Single channel - original and FeatureMarked speech signal .	188
7.12	Multi-channel - original and FeatureMarked speech signal .	189
7.13	Average recovery rate	193
8.1	Amplitude-frequency plot	198
8.2	Walsh spectrum	198
8.3	Sample QR code	200
8.4	Audio segment and subsegment	204
8.5	Construction of embedding information	204
8.6	Arnold transformed QR code	205
8.7	QR code embedding scheme	205
8.8	Walsh code detection	210
8.9	Watermark extraction scheme	212
8.10	Sample 1 - QR code	216
8.11	Sample 2 - encrypted QR code	216

8.12	Single channel - original and FeatureMarked speech signal .	217
8.13	Multi-channel - original and FeatureMarked speech signal .	218
8.14	Average recovery rate	222

List of Tables

3.1	Performance comparison	81
3.2	Performance comparison (contd..)	82
3.3	Performance comparison (contd..)	83
3.4	Advantages and disadvantages of watermarking schemes (contd..)	84
3.5	Advantages and disadvantages of watermarking schemes (contd..)	85
4.1	MFCC values	103
4.2	Spectral flux values	104
4.3	Spectral roll-off values	105
4.4	Centroid values	106
4.5	Entropy values	107
4.6	Energy values	109
4.7	Zero-crossing values	110
5.1	Types of inputs (420 inputs signals of 10 members)	126
5.2	Types of speech files (5 male and 5 female speakers)	126
5.3	Feature selection	139
5.4	Classification accuracy for single features	139
5.5	Classification accuracy for a combination of features	140
5.6	Classification accuracy for a combination of features	140

6.1	5-Point grade scale	163
6.2	Imperceptibility criteria	163
6.3	Common signal manipulations	164
6.4	Desynchronization attacks	165
6.5	Robustness test for signal manipulations (in BER×100%) .	166
6.6	Robustness test for signal manipulations (in BER×100%) .	166
6.7	Robustness test for desynchronization attacks (in BER×100%)	166
6.8	Robustness test for signal manipulations (in BER×100%) .	167
6.9	Robustness test for signal manipulations (in BER×100%) .	167
6.10	Robustness test for desynchronization attacks (in BER×100%)	167
7.1	Imperceptibility criteria	189
7.2	Robustness test for signal manipulations (in BER×100%) .	191
7.3	Robustness test for signal manipulations (in BER×100%) .	191
7.4	Robustness test for desynchronization attacks (in BER×100%)	191
7.5	Robustness test for signal manipulations (in BER×100%) .	192
7.6	Robustness test for signal manipulations (in BER×100%) .	192
7.7	Robustness test for desynchronization attacks (in BER×100%)	192
8.1	Imperceptibility criteria	219
8.2	Robustness test for signal manipulations (in BER×100%) .	220
8.3	Robustness test for signal manipulations (in BER×100%) .	220
8.4	Robustness test for desynchronization attacks (in BER×100%)	220
8.5	Robustness test for signal manipulations (in BER×100%) .	221
8.6	Robustness test for signal manipulations (in BER×100%) .	221
8.7	Robustness test for desynchronization attacks (in BER×100%)	221
9.1	Existing Watermarking Schemes	227
9.2	Proposed Watermarking Schemes	228

Abstract

Presently different audio watermarking methods are available; most of them inclined towards copyright protection and copy protection. This is the key motive for the notion to develop a speaker verification scheme that guarantees non-repudiation services and the thesis is its outcome.

The research presented in this thesis scrutinizes the field of audio watermarking and the outcome is a speaker verification scheme that is proficient in addressing issues allied to non-repudiation to a great extent. This work aimed in developing novel audio watermarking schemes utilizing the fundamental ideas of Fast-Fourier Transform (FFT) or Fast Walsh-Hadamard Transform (FWHT). The Mel-Frequency Cepstral Coefficients (MFCC) the best parametric representation of the acoustic signals along with few other key acoustic characteristics is employed in crafting of new schemes. The audio watermark created is entirely dependent to the acoustic features, hence named as FeatureMark and is crucial in this work.

In any watermarking scheme, the quality of the extracted watermark depends exclusively on the pre-processing action and in this work framing and

windowing techniques are involved. The theme non-repudiation provides immense significance in the audio watermarking schemes proposed in this work. Modification of the signal spectrum is achieved in a variety of ways by selecting appropriate FFT/FWHT coefficients and the watermarking schemes were evaluated for imperceptibility, robustness and capacity characteristics. The proposed schemes are unequivocally effective in terms of maintaining the sound quality, retrieving the embedded FeatureMark and in terms of the capacity to hold the mark bits.

Robust nature of these marking schemes is achieved with the help of synchronization codes such as Barker Code with FFT based FeatureMarking scheme and Walsh Code with FWHT based FeatureMarking scheme. Another important feature associated with this scheme is the employment of an encryption scheme towards the preparation of its FeatureMark that scrambles the signal features that helps to keep the signal features unrevealed.

A comparative study with the existing watermarking schemes and the experiments to evaluate imperceptibility, robustness and capacity tests guarantee that the proposed schemes can be baselined as efficient audio watermarking schemes. The four new digital audio watermarking algorithms in terms of their performance are remarkable thereby opening more opportunities for further research.

Chapter 1

Introduction

1.1 Motivation

Advances in digital technology have led to widespread use of digital communication in various areas including government, legal, banking and military services. This in turn has increased the reproduction and re-transmission of multimedia data through both legal and illegal channels. However, the illegal usage of digital media causes a serious threat to the content owner's authority or proprietary right. Thus, today's information driven society places utmost importance on authenticating the information that is sent across various communication channels. In the case of digital audio communication schemes these disputes may be the denial of authorship of the speech signal, denial of sending or receiving the signal, denial of time of occurrence etc. Incorporating non-repudiation services in this context guarantees the occurrence of a particular event, the time of occurrence as well as the parties and the corresponding information associated with the event.

Typically, a non-repudiation service should produce cryptographic evi-

dence that guarantee dispute resolution. In other terms, the service should hold relevant information that can achieve the goals against denying their presence or participation. Development of a non-repudiation service should have a service request, in the sense that, the parties involved should agree to utilize the service as well as to generate necessary evidence to support their presence. Evidence of this scheme should be transferred to the other party for the purpose of verification and storage. Separate evidence should be available for the originator as well as the recipient by considering the fact that, any one will not gain any extra benefit from this service and to ensure that the concept of fairness is applied. Timeliness and confidentiality are the other features of a non-repudiation service.

Currently most audio watermarking methods available are inclined towards copyright protection and copy protection. This is the key motive for the notion to develop a speaker verification scheme that guarantees non-repudiation services and the thesis is its outcome. Developing a non-repudiating voice authentication scheme is a challenging task in the context of audio watermarking. Our aim is to suggest a digital audio watermarking scheme that ensures authorized and legal use of digital communication, copyright protection, copy protection etc. that helps to prevent such disputes. Audio watermarking is the term coined to represent the insertion of a signal, image or text of known information in an audio signal in an imperceptible form. The embedded watermark should be robust to any signal manipulations and can be unambiguously retrieved at the other end.

1.2 Problem Statement

Evolution in digital technology led to widespread use of digital communication and illegal usage of digital media causes a serious threat to the content

owner's authority or proprietary right. Recent copyright infringements in digital communication make us believe that the stronger analytical tools and methods need to be researched on.

In order to combat this malicious usage of digital audio communication we need to:

- Understand the existing audio watermarking schemes especially that are proposed towards Intellectual Property Rights (IPR);
- Understand some of the best practices in existing watermarking schemes;
- Identify a differentiator for the new schemes which in turn results in developing signal dependent watermarks;
- Classify the key acoustic characteristics that facilitate to uniquely identify the speaker by creating dedicated FeatureMarks;

1.3 Objectives of the Proposed Study

- Extract the key signal contingent features associated with the acoustic signals;
- Identify appropriate features that enable us to identify the speaker by employing artificial neural networks (ANN), k-nearest neighbors (k-NN) and support vector machine (SVM) classifiers;
- Craft the signal reliant watermark using the appropriate extracted features;
- Embed the new watermark or FeatureMark using Fast Fourier Transform (FFT);

- Embed the new watermark or FeatureMark using Fast Walsh-Hadamard Transform (FWHT);
- Evaluation of the proposed schemes in terms of imperceptibility, robustness and capacity;
- Demonstrate speaker authentication as well as non-repudiation competency of the scheme.

1.4 Scope of the Work

The work introduces three novel but diverse voice signal authentication schemes that assure non repudiation by utilizing the key acoustic signal features towards the preparation of the watermark. As part of this research ANN, k-NN and SVM classifiers are employed in tagging the appropriate acoustic features in the new FeatureMark. Acoustic characteristics such as Mel-frequency cepstral coefficients (MFCC), spectral roll-off, spectral flux, spectral centroid, zero-cross rate, energy entropy and short-time energy are vital to this research. This research also illustrate the watermark embedding algorithms which is central to this research. Experiments to determine the behaviors of the proposed schemes in terms of imperceptibility, robustness and capacity is also a component of this work. Main idea behind this work; realization of a non-repudiation service; is achieved in such a way that the speaker in the communicating group cannot subsequently deny their participation in the communication due to the signal-dependent dynamic watermark.

- Scope
 - Determination of apt audio features by conducting speaker recognition using different classifiers

-
- Component-based FeatureMarking system with FFT, Barker code and data matrix
 - Component-based FeatureMarking system with FWHT, Walsh code and quick response (QR) Code
 - Evaluation of the FeatureMark strength
 1. Transparency Tests
 2. Robustness Tests
 3. Capacity Tests

The watermark technique introduced allows tracking the spread of illicit copies but does not do anything to limit the number of copies allowed or control its dissemination through computer networks or other digital media such as compact disks. This research doesn't gaze on the impact of Human Language or impact of mimic sounds on the proposed watermark.

1.5 System Framework

The research involves iteration of steps from collection of acoustic samples from diverse speakers to the detection of embedded FeatureMark. The initial step is to collect different speech signals from people. Next step involves pre-processing of speech signal using the framing and windowing methods. Pre-processed signals are given into the feature extraction module. Once the features are extracted and stored in database, classification module starts functioning to determine some of the apt features that can identify speakers uniquely or in combination with other features.

The actual watermarking algorithm starts its functions only at this step and it needs the watermark developed using the extracted signal features as input. In order to prepare the signal dependent watermark, also termed

as FeatureMark in the suggested schemes; online data-code generators are employed. In some cases, a synchronization code is also generated that could guarantee robustness of the watermarking schemes. FeatureMark embedding is performed by either transforming the signal using FFT or FWHT transforms. Embedded signal is inverse transformed and send to the other end.

Overview of the proposed schemes is presented in the following figures - figure 1.1 and figure 1.2.

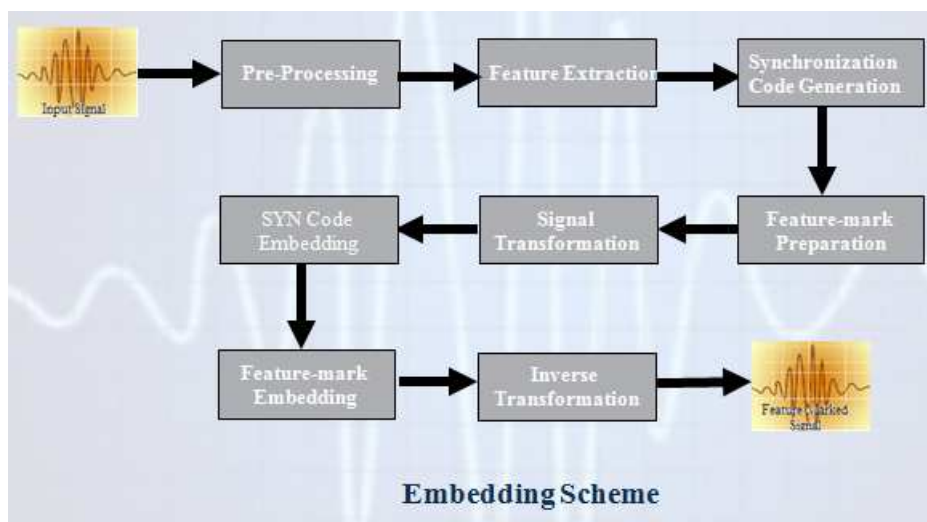


Figure 1.1: Watermark embedding scheme

At the receiving end, presence of watermark is confirmed by performing proper signal transforms. Once the watermark has been detected, it should be extracted to confirm the authenticity of the signal. This guarantees the proof of ownership as the watermark itself holds information about the speakers. The watermark can be enhanced to hold the location, date and

time of event of the communication.

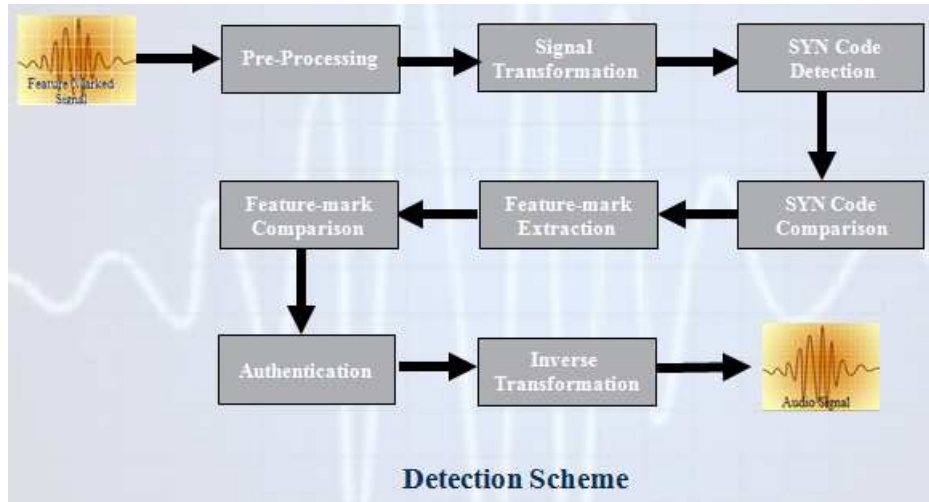


Figure 1.2: Watermark detecting scheme

1.6 Thesis Contributions

This dissertation contributes to the area of pure experimental computer science and introduces novel thinking and techniques to the fields of audio watermarking. The primary objective of this dissertation is to test the hypothesis that:

- digital communication require and should benefit from novel non-repudiation service designed to exploit the acoustic characteristics of the parties involved in the communication.

It should be eminent that it is not possible to formally prove the rightness or falsehood of this hypothesis. Instead, this dissertation is limited to

providing strong evidence for or against its validity. It does so by introducing three new FeatureMarking techniques and revelation of its experimental results. Proposed schemes were able to showcase improvement in terms of imperceptibility, robustness and capacity.

Major contributions of this work involve suggestion of a model for the generation of signal dependent dynamic watermarks that assures authenticity. Through this research three different audio watermarking schemes are offered in which the first one is an acoustic authentication scheme with FFT and barcode as the watermark, second one is a varying audio watermarking system with FFT and data matrix code as the watermark and the final scheme works with FWHT and QR code as the watermark that supports non-repudiation services.

- Proposed a model for the generation of signal dependent dynamic watermarks that assures authenticity of the signal rather than using the regular static ones.
- A speech signal authentication scheme is proposed that works with FFT and uses barcode as the watermark.
- A varying audio watermarking scheme is suggested by employing data-matrix code as the watermark and uses FFT for the marking/unmarking schemes.
- Another method that supports non-repudiation services to a great extent are implemented with the help of FWHT and QR code as the watermark.
- An encryption scheme is suggested that adds one more layer of security to the signal dependent dynamic watermark.

1.6.1 List of Research Papers

As part of the research work various papers were presented and published in peer reviewed International Journals as well as in Conference proceedings. They are listed below:

- Remya A R, A Sreekumar and Supriya M. H. “Comprehensive Non-repudiate Speech Communication Involving Geo-tagged FeatureMark”, Transactions on Engineering Technologies - World Congress on Engineering and Computer Science 2014, Springer Book. *Accepted*
- Remya A R, A Sreekumar. “User Authentication Scheme Based on Fast-Walsh Hadamard Transform”, IEEE Digital Explore Library - 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT], Noorul Islam University (NIUEE), Thuckalay. 978-1-4799-7074-2/15 ©2015 IEEE. *Accepted*
- Remya A R, A Sreekumar. “An FWHT Based FeatureMarking Scheme for Non-repudiate Speech Communication ”, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2014, 22-24 October, 2014, San Francisco, USA. ISBN: 978-988-19252-0-6. *Accepted*
- Remya, A. R., et al. “An Improved Non-Repudiate Scheme - FeatureMarking Voice Signal Communication.”International Journal of Computer Network & Information Security 6.2 (2014)
- Remya, A. R., M. H. Supriya, and A. Sreekumar. “A Novel Non-repudiate Scheme with Voice FeatureMarking.”Computational Intelligence, Cyber Security and Computational Models. Springer India, 2014. 183-194.

- Remya A R, A Sreekumar, “Voice Signal Authentication with Data matrix Code as the Watermark ”, International Journal of Computer Networks and Security, ISSN: 2051-6878, Vol.23, Issue.2, 2013
- Remya A R, A Sreekumar, “Authenticating Voice Communication with Barcode as the Watermark”, International Journal of Computer Science and Information Technologies, Vol. 4 (4) , 2013, 560 - 563
- Remya A R, A Sreekumar, “An Inductive Approach to the Knack of Steganology”, International Journal of Computer Applications (0975 - 8887), Volume 72 - No.15, June 2013
- Remya A R, A Sreekumar, “A Review on Indian Language Steganography ”, CSI Digital Resource Center, National Conference on Indian Language Computing NCILC’13

1.7 Thesis Outline

The thesis is divided into nine chapters and a brief description of each chapter is given below.

Chapter 1 is a general introduction on the importance of watermarking especially audio watermarking. The chapter concludes the significance of the present work.

Chapter 2 is a documentation of the background study conducted to understand the audio signals, an overview of human auditory system, the frequency component analysis of signals and finally the concept of watermarking and its evaluation strategies.

Chapter 3 comprises of a brief description of the existing works that are proposed in the field of audio watermarking. The existing schemes

are mainly classified under three categories such as time domain based algorithms, transform domain based algorithms and hybrid algorithms.

Chapter 4 focuses on the collection of speech data and how pre-processing is done to improve the result. Short-term processing of the signal manipulates the sound inputs appropriately and helps in improving the results of analysis and synthesis. It also guarantees a better quality for the extracted watermark. Feature extraction is another important step described in this chapter where some of the computational characteristics of speech signals are mined for later investigation. Features are extracted using the program code in Matlab by employing the FFT on the time domain signals. Features selected for this study includes physical features such as Mel-frequency cepstral coefficients (MFCC), spectral roll-off, spectral flux, spectral centroid, zero-cross rate, short-time energy, energy entropy and fundamental frequency, that directly correspond to the computational characteristics of the signal and not related to the perceptual characteristics.

Chapter 5 is dealing with the identification of exact features (from the features that we have chosen) that helps in speaker authentication to a great extent. This is achieved by employing three main classifiers such as ANN, k-NN and SVM for individual feature sets as well as different combinations of feature sets. This speaker recognition module reveals that MFCCs itself can help in identifying the speakers participated in the communication system. Different combinations of signal features such as MFCCs, spectral roll-off, zero-cross rate, spectral flux as well as spectral centroid are opted towards the creation of its signal dependent watermark.

Chapter 6 includes the first scheme that we have proposed towards authenticating each member who has participated in the communication system. This scheme works in the transform domain of an audio signal by employing FFT towards embedding and detection schemes. The pre-

pared watermark is a data code and employs Arnold/Anti-Arnold transform in embedding/extracting schemes for scrambling/de-scrambling the watermark. This two-dimensional watermark is transformed into a one-dimensional sequence of 1s and 0s (binary digits) to embed into the audio signal. To evaluate the efficiency of this method, subjective listening tests were conducted which demonstrates the transparency criteria. Robustness tests confirmed the strength against common signal manipulations and de-synchronization attacks and finally the capacity of this scheme was evaluated.

Chapter 7 demonstrates the second scheme which is a variation on the previous method with utilization of a 13-bit Barker code as synchronization code and a data matrix code as watermark in the embedding module. Embedding a synchronization code helps to locate the position of watermark in the modified signal which in turn reduces computational time of the system. FeatureMark embedding and thus its detection is achieved by transforming the signal using FFT. In this scheme also, efficiency tests were conducted to evaluate the transparency, robustness and capacity characteristics.

Chapter 8 introduces the third scheme that works with FWHT. In this scheme, a 16-bit Walsh code is generated and employed as the synchronization code and QR code is treated as its FeatureMark. A variation of this scheme is also suggested which incorporates an encryption scheme in the development of signal dependent watermark. Efficiency of this scheme is also tested by employing subjective listening tests that confirm transparency characteristics. Robustness tests are conducted to find out how the system is robust against common signal manipulations and de-synchronization attacks. Then capacity test is performed to identify the capacity of the proposed watermarking scheme.

Chapter 9 is summary of the work, where important conclusions such as the use of voice signal features, its classification and FeatureMarking towards the development of a secure, robust, voice authentication scheme that helps in guaranteeing the non-repudiation services are highlighted. A comparative study with the existing watermarking schemes is also presented. Towards the end of this chapter the future scope of the proposed works are given.

List of notations, abbreviations, publications, references and index are given at the end of this book.

1.8 Summary

The introductory chapter gives an idea about the work that we have done and the thesis contributions. With the existing audio watermarking algorithms we can guarantee copyright protection, copy protection and ownership to a great extent. But any of these schemes does not employ a signal dependent, dynamic watermark and this is the advantage that can be availed in the suggested schemes. And moreover embedding this FeatureMark helps in guaranteeing the ownership as well as non-repudiation service in a straight way.

Chapter 2

An Overview of Signal Processing & Watermarking

2.1 Introduction

The focus of this chapter is to provide a brief idea on the concepts and techniques used in the proposed study. This includes a brief description on audio signals, an overview of human auditory system (HAS), the frequency component analysis of signals, the concept of watermarking and its evaluation strategies. The theoretical background is explained in seven sections detailed as follows.

2.2 Signals and Systems

In this present world, we are coming across different kinds of signals in various forms. Some of the signals are natural and others are man-made. In this, some are necessary such as speech, some are pleasant such as music

and many are unwanted or unnecessary in a given context. In an engineering context, signals are carriers of both useful and unwanted information. From this mix of conflicting information, useful information can be extracted or enhanced by signal processing. Thus, signal processing can be defined as an operation designed for the extraction, enhancement, storage or transmission of useful information. The distinction between useful and unwanted information which depends on the context can be subjective or objective and thus signal processing is application dependent.

Most of the signals that we encounter in practice are analog signals which are signals that vary in time and amplitude and their processing using electrical networks containing active and passive circuit elements are termed as analog signal processing (ASP). The main drawback of ASP is its limited scope for performing complicated signal processing applications. Therefore, one needs to convert analog signals into a form suitable for digital hardware and termed as digital signals. These signals can take one of the finite numbers of values at specific instances in time and therefore be represented by binary numbers or bits and their processing is termed as digital signal processing (DSP).

Signals bear exact information that the DSP system is trying to interpret. The main purpose of a DSP system is to provide the best approach to analyze and estimate the information content of the signal. Two important categories of DSP are signal analysis and signal filtering and are depicted using the following figure 2.1 [Ingle and Proakis 2011a]:

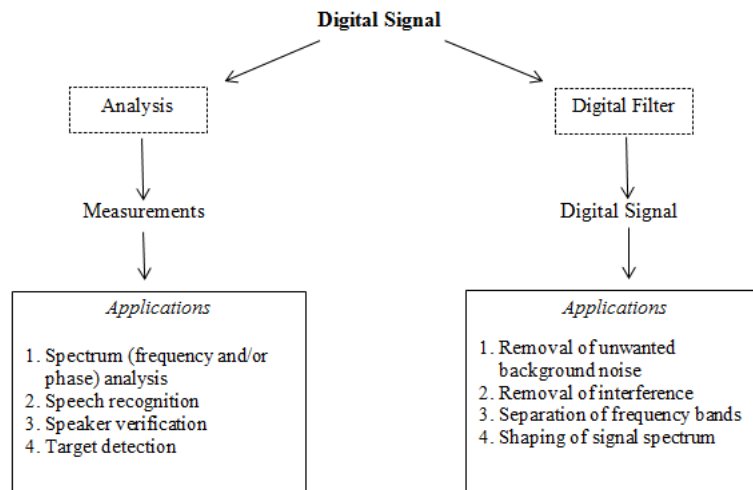


Figure 2.1: Digital signal processing

Signal analysis is the term coined to demonstrate the process of defining and quantifying all signal characteristics for the application being processed [DeFatta, Lucas, and Hodgkiss 1995; Palani and Kalaiyarasi 2011; Ingle and Proakis 2011b; Leis 2002; WolfRam 2011; S 2012; Goldsmiths 2001].

2.3 Audio Signals

2.3.1 Description

An audio signal is a representation of sound, usually in decibels and rarely as voltages [MusicTech 2011]. An exciting human sensory capability is the hearing system [Plack 2007]. Audio frequency range denotes the limits of human hearing and is in the range of 20 - 20,000 Hz [Encyclopedia 2013] and intensity range of 120 dB. A digital audio signal is the result of suitable

sampling and quantization performed on an audio signal with a sampling rate of 44,100 Hz. Audio signal processing, sometimes referred to as audio processing, is the intentional alteration of auditory signals or sound, often through an audio effect or effects unit. As audio signals may be electronically represented in either digital or analog format, signal processing may occur in either domain. Analog processors operate directly on the electrical signal, while digital processors operate mathematically on the digital representation of that signal [Encyclopedia 2013].

As described in [McLoughlin 2009], sound can either be created through the speech production mechanism or as heard by a machine or human. In purely physical terms, sound is a longitudinal wave which travels through air or a transverse wave in some other media due to the vibration of molecules. In air, sound is transmitted as a pressure variation between high and low pressure, with the rate of pressure variation from low to high, to low again, determining the frequency. The degree of pressure variation (namely the difference between high and low) determines the amplitude.

2.3.2 Characteristics

Hearing sense serves as one of the gateways to the external environment by providing us information regarding the locations and characteristics of sound producing objects. An important characteristic of HAS is the ability to process the complex sound mixture received by the ears and form high-level of abstractions of the environment by the analysis and grouping of measured sensory inputs. Auditory scene analysis is the term coined to demonstrate the process of achieving the segregation and identification of sources from the received composite acoustic signal. The concept of sound source separation and classification in its reality comes in applications including speech recognition, automatic music transcription, multimedia data

search and retrieval and audio watermarking. In all these cases, the audio signal must be processed based on the signal models which may be drawn from sound production, sound perception and cognition. Real-time applications of digital audio signal processing include audio data compression, synthesis of audio effects, audio classification, audio steganography and audio watermarking. Unlike images, audio records can only be listened sequentially; good indexing is valuable for effective retrieval. Listening to audio clips can actually help to navigate audio visual materials more easily than the viewing of video scenes.

The properties of an audio event can be categorized as temporal or spectral properties. The temporal properties refer to the duration of the sound and any amplitude modulations; the spectral properties of the sound refer to its frequency components and their relative strengths.

Audio waveforms can be categorized as periodic or aperiodic waveforms. Complex tones comprising of fundamental frequency and multiples of the fundamental frequency are grouped under the periodic waveforms. Non-harmonically related sine tones or frequency shaped noise forms the aperiodic waveforms. As discussed in [Prasad and Prasanna 2008], sound signals are basically physical stimuli that are processed by the auditory system to evoke psychological sensations in the brain. It is appropriate that the salient acoustical properties of a sound be the ones that are important to the human perception and recognition of the sound. Studies on hearing perception have been started since 1870s at the time of Helmholtz. The perceptual attributes of sound waves are pitch, loudness, subjective duration and timbre.

HAS is known to carry out the frequency analysis of sounds to feed the higher level cognitive functions. Audio signals are represented in terms of a joint description of time and frequency because both spectral and tempo-

ral properties are relevant to the perception and cognition of sound. Audio signals are non-stationary in nature and the analysis of each signal assumes that the signal properties change slowly with respect to time. Signal characteristics are estimated based on the time center of the each short-windowed segment and the analysis is repeated at uniformly spaced intervals of time. Short-time analysis is the term coined to represent the method of estimating the parameters of a time-varying signal and the obtained features are termed as the short-time parameters which relate to an underlying signal model.

2.3.3 Representation

The acoustic properties of sound events can be visualized in a time-frequency “image” of the acoustic signal. Human auditory perception starts with the frequency analysis of the sound in the cochlea. The time-frequency analysis of sound is therefore a natural starting point for machine-based segmentation and classification. Two important audio signal representations are spectrogram and auditory representation that help to visualize the spectro-temporal properties of sound waves. First one is based on adapting the Fourier transform to time-varying analysis and the second one incorporates the knowledge of hearing perception to emphasize perceptually salient characteristics of the signal.

Spectrogram

Incorporating Fourier transforms in the spectral analysis of a time-domain signal produces a pair of real-valued functions of frequency called the amplitude/magnitude spectrum and the phase spectrum. Audio signals are

segmented and the time-varying characteristics of each segment are analyzed using the Fourier transforms at short successive intervals. That is, spectrogram is the visual representation of the time-frequency analysis of individual acoustic frames that may overlap in time and frequency. The duration of the analysis window dictates the trade-off between the frequency resolution and time resolution of steady-state content and time-varying events respectively.

Auditory Representations

The way in which the time-varying signal is perceived by the human can be better visualized with the auditory representations. By this way, the perceptually salient features of the audio signals are more directly evident than in the spectrogram. In other words, spectrogram visualizes the spectro-temporal properties according to the physical intensity levels whereas auditory phenomena take care of the human ear's sensitivity to different components. The main components that affect human audible range include the hearing ability in the low, middle and high frequency regions, signal intensity versus perceived loudness, decreasing frequency resolution versus increasing frequency.

2.3.4 Features

Auditory signal representations discussed above are good for visualization of the audio content but they have high dimensionality and make them unsuitable in applications such as classification, information hiding etc. This in turn results in the extraction of low-dimensional features that holds only the most important and distinctive characteristics of each signal. Linear transformation of a spectrogram proposed in MPEG-7 , the audiovi-

sual content standard presented in [Martinez 2002, Xiong et al. 2003] extracts reduced-dimension and de-correlated spectral vectors. As discussed in [Prasad and Prasanna 2008], features are designed with the help of salient signal characteristics in terms of signal production or perception. Main aim is to find features that are invariant to irrelevant transformations and have good discriminative power across classes. Feature values correspond to the numerical representation of acoustic signals are used to characterize the audio segment. Features can be either physical and perceptual or static and dynamic.

Physical Features

Physical features are directly related to the computable characteristics of time-domain signals and not related to the human perception. These characterize the low-level or reduced-dimension parameters and thus stand for specific temporal and spectral properties of the signal. But some perceptually motivated features are also classified under physical features since they can be extracted directly from the audio waveform amplitudes or the short-time spectral values. [Prasad and Prasanna 2008] represents the audio signal analysis for the r th frame as below:

$$\begin{cases} X_r[n] & X_r[k] \text{ at frequency } f[k], \\ n = 1 \dots N & k = 1 \dots N. \end{cases} \quad (2.1)$$

where, the subindex “ r ” indicates the current frame so that $x_r[n]$ are the samples of the N -length data segment (which is possibly multiplied by a window function) corresponding to the current frame.

Mainly used acoustic features in our works are discussed below:

- Mel-Frequency Cepstral Coefficients [MFCC]

Mel-frequency cepstral coefficients introduced by Davis and Mermelstein in 1980's are treated as the best parametric representation of the acoustic signals employed in the recognition of speakers and have been the state-of-the-art ever since. Mel-scale relates perceived frequency or pitch of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely to what humans hear. MFCCs are based on a linear cosine transform of a log power spectrum on a non-linear Mel-scale of frequency. MFCCs are treated as the best for speech or speaker recognition systems because it takes human sensitivity with respect to frequencies into consideration. [Gopalan 2005a; Gopalan 2005b; Kraetzer and Dittmann 2007] demonstrates audio watermarking as well as audio steganographic techniques in the cepstral coefficients.

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125 \ln(1 + f/700) \quad (2.2)$$

To go from Mels back to frequency:

$$M^{-1}(m) = 700(\exp(m/1125) - 1) \quad (2.3)$$

According to [Prasad and Prasanna 2008], in order to compute the MFCC, the windowed audio data frame is transformed by a DFT. Mel-scale filter bank is then applied in the frequency domain and the power within each sub-band is computed by squaring and summing

the spectral magnitudes within bands. The Mel-frequency scale, a perceptual scale like the critical band scale, is linear below 1 kHz and logarithmic above this frequency. Finally the logarithm of the band-wise power values are taken and de-correlated by applying a DCT to obtain the cepstral coefficients. The log transformation serves to de-convolve multiplicative components of the spectrum such as the source and filter transfer function. The de-correlation results in most of the energy being concentrated in a few cepstral coefficients. For instance, in 16 kHz sampled speech, 13 low-order MFCCs are adequate to represent the spectral envelope across phonemes [Lyons 2009; Encyclopedia 2013].

A related feature is the cepstral residual computed as the difference between the signal spectrum and the spectrum reconstructed from the prominent low-order cepstral coefficients. The cepstral residual thus provides a measure of the fit of cepstral smoothed spectrum to the spectrum.

- Spectral Flux

The spectral flux also termed as spectral variation is a measure of how quickly the power spectrum varies corresponding to each frames in a short-time window. It can be defined as the squared difference between the normalized magnitudes of successive spectral distributions corresponding to successive signal frames. Thus, spectral flux can be described as the local spectral rate of change of an acoustic signal. Timbre of an audio signal can also be derived from it [Encyclopedia 2013]. A high value of spectral flux stands for a sudden change in the spectral magnitudes and therefore a possible spectral boundary at the r^{th} frame.

Spectral flux can be calculated as follows:

$$F_r = \sum_{k=1}^{\frac{N}{2}} |X_r[k]| - |X_{(r-1)}[k]|^2 \quad (2.4)$$

where $X_r[k]$ represents the normalized magnitudes of spectral distribution corresponding to signal frame X_r .

- Zero-Cross Rate(ZCR)

Zero-cross rate is the key feature used in classifying voice signals or musical sounds. ZCR is calculated for each frame and is defined as the rate of sign changes along a signal [Encyclopedia 2013].

$$ZCR = \frac{1}{T-1} \sum_{t=1}^{T-1} \Pi\{S_t S_{t-1} < 0\} \quad (2.5)$$

where S is a signal of length T and the indicator function $\Pi\{A\}$ is 1 if its argument A is true and 0 otherwise.

- Spectral Centroid

The spectral shape of a frequency spectrum is measured with the spectral centroid. Higher the centroid values, the brighter will be the textures with more high frequencies. This measure characterises a spectrum and can be calculated as weighted mean of the frequencies presented in the signal, determined using a Fourier transform with their magnitudes as weights:

$$C = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)} \quad (2.6)$$

where $X_r[n]$ represents the weighted frequency value or magnitude of binary number n and $f(n)$ represents the center frequency of that binary number.

Centroid represents sharpness of the sound which is related to the high-frequency content of the spectrum. Higher centroid values resemble the spectra in the range of higher frequencies. The effectiveness of centroid measures to describe spectral shape makes it usable in voice signal classification activities [Encyclopedia 2013].

- Spectral Roll-Off

Spectral roll-off point is defined as the N^{th} percentile of the power spectral distribution where N is usually 85% or 95%. The roll-off point is the frequency below which $N\%$ of the magnitude distribution is concentrated. In other words, spectral roll-off demonstrates the frequency below which 85% of the magnitude distribution of the spectrum is concentrated. Both the centroid and spectral roll-off are measures of spectral shape and the spectral roll-off yields higher values for high frequencies or right skewed spectra [Encyclopedia 2013; SOVARRwiki 2012; Datacom 2012].

The roll-off is given by $R_r = f[k]$, where K is the largest bin that satisfies the below equation 2.7 :

$$\sum_{k=1}^K |X_r[k]| \leq 0.85 \sum_{k=1}^{\frac{N}{2}} |X_r[k]| \quad (2.7)$$

- Energy Entropy

Importance of energy entropy criterion is in the context of capturing sudden changes in the energy levels of an audio signal. Each audio

frame is further segmented into sub-windows and the energy-entropy is calculated for each of these windows which have a fixed-duration. For each sub-window i , the normalized energy is calculated, i.e., the sub-window's energy divided by the whole window's energy. Then, the energy entropy is computed for frame j using the following equation 2.8.

$$I_j = - \sum_{i=1 \dots k} \sigma_i^2 \log_2 \sigma_i^2 \quad (2.8)$$

It is summarized that the value of energy entropy is low for frames with large changes in its energy level [Giannakopoulos et al. 2006].

- Short-time Energy

The short-time energy of speech signals reflects the amplitude variation and is calculated using the following equation 2.9 :

$$E_{\hat{k}} = \sum_{m=-\infty}^{\infty} (x(m)w[\hat{k} - m])^2 = \sum_{m=-\infty}^{\infty} (x^2(m)w^2[\hat{k} - m]) \quad (2.9)$$

or can be expressed as follows; which is the long-term definition of signal energy

$$N_j = \sum_{i=1 \dots S} x_i^2 \quad (2.10)$$

In order to reflect the amplitude variations in time (for this a short window is necessary) and considering the need for a low pass filter to provide smoothing, $h(k)$ was chosen to be a hamming window powered by 2. The short-time energy helps to differentiate the voiced speech

from un-voiced speech [Prasad and Prasanna 2008; Giannakopoulos et al. 2006; Anguera 2011; Rabiner 2012].

- Band-level Energy

Representing energy of the time-domain signal within a specified frequency region of the signal spectrum is by means of the band-level energy. As given in [Prasad and Prasanna 2008], it can be computed by the appropriately weighted summation of the power spectrum as follows:

$$E_r = \frac{1}{N} \sum_{k=1}^{\frac{N}{2}} (X_r[k]W[k])^2 \quad (2.11)$$

$W[k]$ is a weighting function with non-zero values over only a finite range of bin indices “k” corresponding to the frequency band of interest. Sudden transitions in the band-level energy indicate a change in the spectral energy distribution, or timbre, of the signal, and aid in audio segmentation. Generally log transformations of energy are used to improve the spread and represent (the perceptually more relevant) relative differences.

- Fundamental Frequency (f_0)

Fundamental frequency, f_0 is measured with respect to the periodicity of the time-domain signal. Or, it can be taken as the frequency of the first harmonic or as the spacing between harmonics of the periodic signal in the signal spectrum.

Perceptual Features

Perceptual features correspond to the subjective perception of the sound and are extracted using auditory models. Thus, human recognition of sound

is based on these features. As described in [Prasad and Prasanna 2008], the psychological sensations evoked by a sound can be broadly categorized as loudness, pitch and timbre. Loudness and pitch can be ordered on a magnitude scale of low to high whereas timbre is based on several sensations that serve to distinguish different sounds of identical loudness and pitch. Numerical representations of short-time perceptual features are evaluated using computational models for each audio segment. Loudness and pitch with their temporal fluctuations are the common perceptual features of a time-domain signal.

- Loudness

Generally the loudness of a sound is related to the amplitude of the sound wave; a wave with bigger variations in pressure generally sounds louder [Nave 2014]. Loudness of an acoustic signal is correlated with the duration and spectrum of the sound signal as well as with the sound intensity which corresponds to the energy per second reaching a given area. In physiological terms, the perceived loudness is determined by the sum total of the auditory neural activity elicited by the sound. Loudness scales nonlinearly with sound intensity. Corresponding to this, loudness computation models obtain loudness by summing the contributions of critical band filters raised to a compressive power [Prasad and Prasanna 2008]. Salient aspects of loudness perception captured by loudness models are the nonlinear scaling of loudness with intensity, frequency dependence of loudness and the additive of loudness across spectrally separated components.

- Pitch

The main component that gives us the perception of the pitch of a musical note is the fundamental frequency, measured in Hertz. Thus

it can be said that, even though pitch is a perceptual attribute, it is closely correlated with the physical attribute of the fundamental frequency (f_0). Subjective pitch changes are related to the logarithm of f_0 , so that a constant pitch change in music refers to a constant ratio of fundamental frequencies. Most pitch detection algorithms (PDAs) extract f_0 from the acoustic signal, i.e. they are based on measuring the periodicity of the signal via the repetition rate of specific temporal features or by detecting the harmonic structure of its spectrum. A challenging problem for PDAs is the pitch detection of a voice when multiple sound sources are present as occurs in polyphonic music.

- Timbre

If a trumpet and a clarinet play the same note, the difference between the two instruments can easily be identified. Likewise, different voices sound different even when singing the same note. It is understood that if they are playing or singing the same pitch, fundamental frequency is same for both, so it is not the pitch that enables us to tell the difference. These differences in the quality of the pitch are called timbre and depend on the actual shape of the wave which in turn depends on the other frequencies present and their phases [Nave 2014].

And it is understood that pitch is primarily determined by the fundamental frequency of a note. Perceived loudness is related to the intensity or energy per time per area arriving at the ear. Timbre is the quality of a musical note and is related to the other frequencies present [Nave 2014].

2.4 Overview of Human Auditory System

Watermarking of audio signals is more challenging compared to the watermarking of images or video sequences, due to wider dynamic range of the HAS in comparison with human visual system (HVS). The HAS perceives sounds over a range of power greater than 109:1 and a range of frequencies greater than 103:1. The sensitivity of the HAS to the additive white Gaussian noise (AWGN) is high as well; this noise in a sound file can be detected as low as 70 dB below ambient level. On the other hand, opposite to its large dynamic range, HAS contains a fairly small differential range, i.e. loud sounds generally tend to mask out weaker sounds. Additionally, HAS is insensitive to a constant relative phase shift in a stationary audio signal and some spectral distortions interprets as natural, perceptually non-annoying ones. Auditory perception is based on the critical band analysis in the inner ear where a frequency-to-location transformation takes place along the basilar membrane. The power spectra of the received sounds are not represented on a linear frequency scale but on limited frequency bands called critical bands. The auditory system is usually modeled as a band-pass filter bank, consisting of strongly overlapping band-pass filters with band-widths around 100 Hz for bands with a central frequency below 500 Hz and up to 5000 Hz for bands placed at high frequencies. If the highest frequency is limited to 24000 Hz, 26 critical bands have to be taken into account.

Two properties of the HAS dominantly used in watermarking algorithms are frequency (simultaneous) masking and temporal masking. The concept using the perceptual holes of the HAS is taken from wideband audio coding (e.g. MPEG compression 1, layer 3, usually called mp3). In the compression algorithms, the holes are used in order to decrease the amount

of the bits needed to encode audio signal without causing a perceptual distortion to the coded audio. On the other hand, in the information hiding scenarios, masking properties are used to embed additional bits into an existing bit stream, again without generating audible noise in the audio sequence used for data hiding [Nedeljko 2004].

2.4.1 Frequency Masking

Frequency (simultaneous) masking is a frequency domain phenomenon where a low level signal, e.g. a pure tone (the maskee), can be made inaudible (masked) by a simultaneously appearing stronger signal (the masker), e.g. a narrow band noise, if the masker and maskee are close enough to each other in frequency. A masking threshold can be derived below which any signal will not be audible. The masking threshold depends on the masker and on the characteristics of the masker and maskee (narrow band noise or pure tone). The slope of the masking threshold is steeper toward lower frequencies; in other words, higher frequencies tend to be more easily masked than lower frequencies. It should be pointed out that the distance between masking level and masking threshold is smaller in noise-masks-tone experiments than in tone-masks-noise experiments due to HAS's sensitivity towards additive noise. Without a masker, a signal is inaudible if its SPL is below the threshold in quiet, which depends on frequency and covers a dynamic range of more than 70 dB [Nedeljko 2004].

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

Let $SNR(m)$ be the signal-to-noise ratio resulting from watermark insertion in the critical band m ; the perceivable distortion in a given sub-band is then measured by the noise to mask ratio:

$$NMR(m) = SMR - SNR(m) \quad (2.12)$$

The noise-to-mask ratio, $NMR(m)$ expresses the difference between the watermark noise in a given critical band and the level where a distortion may just become audible; its value in dB should be negative. This description is the case of masking by only one masker. If the source signal consists of many simultaneous maskers, a global masking threshold can be computed that describes the threshold of just noticeable distortion (JND) as a function of frequency. The calculation of the global masking threshold is based on the high resolution short-term amplitude spectrum of the audio signal, sufficient for critical band-based analysis and is usually performed using 1024 samples in FFT domain. In a first step, all the individual masking thresholds are determined, depending on the signal level, type of masker (tone or noise) and frequency range.

After that, the global masking threshold is determined by adding all individual masking thresholds and the threshold in quiet. The effects of the masking reaching over the limits of a critical band must be included in the calculation as well. Finally, the global signal-to-noise ratio is determined as the ratio of the maximum of the signal power and the global masking threshold.

2.4.2 Temporal Masking

In addition to frequency masking, two phenomena of HAS in the time-domain also play an important role in human auditory perception. Those

are pre-masking and post-masking in time [Nedeljko 2004].

The temporal masking effects appear before and after a masking signal have been switched on and off, respectively. The duration of the pre-masking is significantly less than one-tenth that of the post-masking, which is in the interval of 50 to 200 milliseconds. Both pre- and post-masking have been exploited in the MPEG audio compression algorithm and several audio watermarking methods.

2.5 Speech and Audio Signal Processing

A speech or audio signal can be represented as a graph of instantaneous amplitude of the pressure wave as converted to an electrical voltage, versus time. Obtained voltage waveform of a speech signal are sampled at a constant sampling rate or sampling frequency. Sampling rate manages the features which can be analyzed from the signal and usually a sufficiently high sample rate is chosen. A common standard is 8000 samples per second for “telephone-quality” audio; 44.1 kHz for high-quality CD audio and a rate of 11025 Hz is often used in personal computer systems. Let T denote the sample period measured in seconds (s) or milliseconds (ms) or microseconds (μ s). The reciprocal, denoted as s is the sample rate and is measured in samples per second or Hz [Williams and Madisetti 1997; Encyclopedia 2013; Leis 2011].

Sample Period = T_s Sample Rate = s Hz

$$f_s = \frac{1}{T} \quad (2.13)$$

The term, zero-order hold (ZOH) is used to describe the fact that the signal is held constant during sampling. In mathematical terms, the

sampling operation is realized as a multiplication of the continuous signal $x(t)$ by a discrete sampling or “railing” function $r(t)$, where

$$r(t) = \begin{cases} 1.0 & , t = nT, \\ 0 & , \textit{Otherwise} \end{cases} \quad (2.14)$$

These sampling impulses are pulses of unit amplitude at exactly the sampling time instants. The sampled function is then

$$x(n) = x(t)r(t) = x(nT) \quad (2.15)$$

Amplitude quantization is the process of representing the real, analog signal by some particular level in terms of a certain N-bit binary representation. But it introduces some error into the system because its limitation to the 2^N discrete level representations instead of the infinite number of levels in the analog signal.

2.6 Frequency Component Analysis of Signals

Determining frequency content of the signals is an important functionality in the area of signal processing. It can be achieved primarily via the Fourier transform, a fundamental tool in digital signal processing [WolfRam 2011; S 2012; Goldsmiths 2001; Mathworks 1984; Oxford 2007; Schwengler 2013; Miranda 2002; Tanyel 2007; Diniz, Da Silva, and Netto 2010].

2.6.1 The Fourier Transform

The Fourier transform is closely related to Fourier series which is an important technique for analyzing the frequency content of a signal. The

fundamental difference is that the requirement to have a periodic signal is now relaxed.

Continuous Fourier Transform

The continuous time Fourier transform converts a signal $x(t)$ in the time domain into its frequency domain counterpart $X(\Omega)$, where Ω is the true frequency in radians per second. Both of these signals are continuous. Then, Fourier transform or continuous-time/continuous-frequency Fourier transform is defined as:

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t} dt \quad (2.16)$$

Its inverse operation can be defined as

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega)e^{j\Omega t} d\Omega \quad (2.17)$$

These two operations are completely reversible, that is, taking any $x(t)$, computing its Fourier transform and then computing the inverse Fourier transform, the result returns the original signal.

Discrete-Time Fourier Transform

The discrete-time Fourier transform works on sampled signals. When a signal $x(t)$ is sampled at time $t = nT$, resulting in the discrete sampled

$x(n)$ and its transform is defined as:

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-jn\omega} \quad (2.18)$$

with t seconds = n samples $\times T \frac{\text{seconds}}{\text{sample}}$

and $\omega \frac{\text{radians}}{\text{sample}} = \Omega \frac{\text{radians}}{\text{second}} \times T \frac{\text{seconds}}{\text{sample}}$

Since the frequency spectrum is still continuous, the inverse DTFT is expressed as

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{jn\omega} d\omega \quad (2.19)$$

Discrete-Fourier Transform

The discrete-time/discrete-frequency Fourier transform, otherwise known as discrete Fourier transform or DFT, performs sampling on time as well as frequency basis and is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jn\omega_k} \quad (2.20)$$

where $\omega_k = \frac{2\pi k}{N}$ = frequency of the k^{th} sinusoid

The discrete basis vectors are now at points on the complex plane, or $1.e^{jn\omega_k}$. Inverse DFT is performed to convert from the frequency domain samples $X(k)$ back to the time domain samples $x(n)$ and is defined as

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k).e^{jn\omega_k} \quad (2.21)$$

Fast-Fourier Transform

Spectral analysis applications often require DFTs in real-time on contiguous sets of input samples. Computation of the DFT for N input sample points requires N^2 complex multiplies and $N^2 - N$ complex additions for N frequency output points. The real-time computational requirements of the DFT are expressed as a function of the DFT input data rate, F .

$$DFTCMPS = NF(\text{Complex multiplies per second}) \quad (2.22)$$

$$DFTCAPS = (N - 1)F(\text{Complex additions per second}) \quad (2.23)$$

The FFT is a fast algorithm for efficient implementation of the DFT where the number of time samples of the input signal N is transformed into N frequency points. The computational requirements of the FFT are expressed as

$$FFTCMPS = \frac{F}{2} \log_2 N \quad (2.24)$$

$$FFTCAPS = F \log_2 N \quad (2.25)$$

By far the most commonly used FFT is the Cooley-Turkey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1 N_2$ into many smaller DFTs of sizes N_1 and N_2 , along with $O(N)$ multiplications by complex roots of unity traditionally called twiddle factors [Gentleman and Sande 1966].

There are other FFT algorithms distinct from Cooley-Turkey. For $N = N_1 N_2$ with co-prime N_1 and N_2 , one can use the Prime-Factor (Good-Thomas) algorithm (PFA), based on the Chinese Remainder Theorem, to factorize the DFT similarly to Cooley-Turkey but without the twiddle factors. The Rader-Brenner algorithm (1976) is a Cooley-Turkey-like factorization but with purely imaginary twiddle factors, reducing multiplications at the cost of increased additions and reduced numerical stability; it was later superseded by the split-radix variant of Cooley-Turkey. Algorithms that recursively factorize the DFT into smaller operations other than DFTs include the Bruun and QFT algorithms. Bruun's algorithm, in particular, is based on interpreting the FFT as a recursive factorization of the polynomial $z^N - 1$, here into real-coefficient polynomials of the form $z^M - 1$ and $z^{2M} + az^M + 1$.

2.6.2 Hadamard Transform

The Hadamard transform also known as the Walsh-Hadamard transform, is a transform that is not based on sinusoidal functions, unlike the other transforms seen so far. Instead, the elements of its transform matrix are either 1 or -1. When $N = 2n$, its transform matrix is defined by the following recursion [Harmuth 1972; Rao and Elliott 1982].

$$H_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.26)$$

$$H_n = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \quad (2.27)$$

From the above equations 2.26, 2.27, it is easy to understand that the Hadamard transform is unitary.

An important aspect of the Hadamard transform is that, since the elements of its transform matrix are only either 1 or -1, the Hadamard transform needs no multiplications for its computation, leading to simple hardware implementations. Because of this fact, the Hadamard transform has been used in digital video schemes in the past, although its compression performance is not as high as that of the DCT [Jain 1989]. Nowadays, with the advent of specialized hardware to compute the DCT, the Hadamard transform is used in digital video only in specific cases. However, one important area of application of the Hadamard transform today is in code-division multiple access (CDMA) systems for mobile communications, where it is employed as channelization code in synchronous communication systems [Stuber 2011].

The Hadamard transform also has a fast algorithm. For example, the matrix H_3 can be factored as [Jain 1989]

$$H_3 = A_8^3 \quad (2.28)$$

where

$$A_8 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad (2.29)$$

Therefore, the number of additions of a fast Hadamard transform algorithm is of the order of $N \log_2 N$. Walsh transforms are the discrete analog of the Fourier transforms. Due to the fact that Walsh functions and trans-

forms are naturally more suited for digital computation, an effort was made to gradually replace the Fourier transform by Walsh-type transforms.

2.7 Watermarking

A digital watermark is the type of latent indicator secretly embedded in a noise-tolerant signal such as audio or image data. It is typically used to identify the ownership or copyright of material. Watermarking is the process of hiding digital information in a carrier signal in order to confirm its authenticity or integrity as well as to show the identity of its owners. The watermark incorporated should be imperceptible to human senses and should not spoil the integrity of the original signal. Since a digital copy of data is the same as the original, digital watermarking is a passive protection tool. This technique simply marks the signal with the data neither it degrades nor controls access to the data thereby by secure the communication.

Audio watermarking initially started as a sub-discipline of digital signal processing, mainly focus on convenient signal processing techniques to embed additional information to audio sequences. This work proposes the investigation of a suitable transform domain method for embedding the watermark and schemes for the imperceptible modification of the host audio. Very recently watermarking has been placed to a stronger theoretical foundation and becoming a more mature discipline with a proper base in both communication modeling and information theory.

2.7.1 General Model of Digital Watermarking

According to [Cvejic 2004; Nedeljko 2004; Cox et al. 2007], the general model of watermarking can be described as: A watermark message ‘m’ is

embedded into the host signal ‘x’ to produce the watermarked signal ‘s’. The embedding process is dependent on the key ‘K’ and must satisfy the perceptual transparency requirement, i.e., the subjective quality difference between ‘x’ and ‘s’ (denoted as embedding distortion d_{emb}) must be below the just noticeable difference threshold. Before the watermark detection and decoding process takes place, ‘s’ is usually intentionally or unintentionally modified. The intentional modifications are usually referred to as attacks; an attack produce attack distortion d_{att} at a perceptually acceptable level. After attacks, a watermark extractor receives attacked signal ‘r’.

The watermark extraction process consists of two sub-processes, first, watermark decoding of a received watermark message ‘m’ using key ‘K’ and second, watermark detection, meaning the hypothesis test between:

- Hypothesis H0: the received data ‘r’ is not watermarked with key ‘K’, and
- Hypothesis H1: the received data ‘r’ is watermarked with key ‘K’

Depending on a watermarking application, the detector performs informed or blind watermark detection. The watermarking algorithms must be designed to endure the worst possible attacks for a given attack distortion d_{att} , which might be even some common signal processing operations (e.g. dynamic compression, low pass filtering etc.).

2.7.2 Statistical Model of Digital Watermarking

In order to properly analyze digital watermarking systems, a stochastic description of the multimedia data is required. The watermarking of data

whose content is perfectly known to the adversary is useless. Any alteration of the host signal could be inverted perfectly, resulting in a trivial watermarking removal. Thus, essential requirements on data being robustly watermarked are that there is enough randomness in the structure of the original data and that quality assessments can be made only in a statistical sense [Eggers and Girod 2002; Nedeljko 2004].

2.7.3 Communication Model of Digital Watermarking

Watermarking, which is also a method of communication, transmits a message from the watermark embedder to the watermark receiver [Nedeljko 2004; Averkiou 2004; Cox et al. 2007].

The communication model itself can also be described in three ways. In the basic model, the cover work is considered purely as noise. In the second model, the cover work is still considered noise, but this noise is provided to the channel encoder as side information. Finally, the third model does not consider the cover work as noise, but rather as a second message that must be transmitted along with the watermark message in a form of multiplexing. In the basic model, regardless of whether it is an informed detector or a blind detector, the embedding process consists of two basic steps. First, the message is mapped into an added pattern, W_a , of the same type and dimension as the cover work, C_o . However, it should be noted that in authentication applications the goal is not to communicate a message but to learn whether and how a work has been modified since a watermark was embedded.

Second model embeds watermarking as communications with side information at the transmitter and thus gain more robustness than the basic model but cannot accommodate all possible embedding algorithms as it restricts the encoded watermark to be independent of the cover work. Because

the un-watermarked cover work, c_o , is obviously known to the embedder, there is no reason to enforce this restriction.

An alternative model considers watermarking as multiplexed communications where instead of considering the cover work as part of the transmission channel, it is transmitted as a second message along with the watermark message in the same signal, c_w . The two messages, c_o and m , will be detected and decoded by two very different receivers: a human being and a watermark detector, respectively.

2.7.4 Geometric Model of Digital Watermarking

To view a watermarking system geometrically, imagine a high-dimensional space in which each point corresponds to one work and this space is referred as media space. Alternatively, when analyzing more complicated algorithms, that is to consider projections or distortions of media space and these spaces are referred as marking spaces. The system can then be viewed in terms of various regions and probability distributions in media or marking space [Cox et al. 2007]. These include the following:

- The distribution of un-watermarked works indicates how likely each work is.
- The region of acceptable fidelity is a region in which all works appear essentially identical to a given cover work.
- The detection region describes the behavior of the detection algorithm.
- The embedding distribution or embedding region describes the effects of an embedding algorithm.

- The distortion distribution indicates how works are likely to be distorted during normal usage.

A watermark embedder is a function that maps a work, a message, and possibly a key into a new work. This is generally a deterministic function such that for any given original work, c_o , message, ‘m’, and key, ‘k’, the embedder always outputs the same watermarked work, c_w .

Watermark detectors are often designed with an explicit notion of marking space, such a detector consists of a two-step process. The first step, watermark extraction, applies one or more pre-processes to the content, such as frequency transforms, filtering, block averaging, geometric or temporal registration, and feature extraction. The result is a vector - a point in marking space - of possibly smaller dimensionality than the original which is the extracted mark. The second step is to determine whether the extracted mark contains a watermark and, if so, decode the embedded message. This usually entails comparing the extracted mark against one or more predefined reference marks (although other methods are possible). This second step can be thought of as a simple watermark detector operating on vectors in marking space.

Thus, it can be summarized that one purpose of the extraction function is to reduce the cost of embedding and detection. A second purpose is to simplify the distribution of un-watermarked works, the region of acceptable fidelity and/or the distortion distribution so that simple watermarking algorithms will perform well.

2.8 Evaluating Watermarking Systems

Most people who deal with watermarking systems need some way of evaluating and comparing them. Those interested in applying watermarking

to an application need to identify the systems that are most appropriate. Those interested in developing new watermarking systems need measures to verify algorithmic improvements. Such measures may also lead to ways of optimizing various properties [Cox et al. 2007].

2.8.1 The Notion of “Best”

Identifying the criteria of determining a robust watermarking system is of extreme importance in the proper evaluation of watermarking schemes. That is, determine what makes one system better than another or what level of performance would be best. An application dependent watermarking should be evaluated based on that particular application. But the case of comparing the new schemes with other existing schemes can be done by evaluating any improvement on any of its properties and can be finalized as an improved scheme. That is, the schemes can be tested against its robustness, imperceptibility or capacity as well as the detection threshold of the embedded watermarks.

2.8.2 Benchmarking

Identifying the appropriate test criteria follows developing the tests cases that are going to perform. All tests of a given system must be performed using the same parameter settings at the embedder and detector (e.g., a constant embedding strength and detection threshold). An early example of such a testing program was the CPTWG’s effort to test watermarks for copy control in DVD recorders [Bell 1999]. For purposes of watermarking research, there has been some interest in the development of a universal benchmark. This benchmark could be used by a researcher to assign a single, scalar “score” to a proposed watermarking system. The score could

then be used to compare it against other systems similarly tested. A proposed benchmark for image watermarking systems [Kutter and Petitcolas 1999] specifies a number of bits of data payload that must be embedded, along with a level of fidelity that must be maintained, as measured by a specific perceptual model. Holding these two properties constant, the robustness of the system to several types of distortion is tested using a program called Stirmark [Petitcolas, Anderson, and Kuhn 1998]. This program applies distortions that have little effect on the perceptual quality of images, but are known to render most watermarks undetectable. The system's performance in these tests is combined into a single score.

Unfortunately, this benchmark can only be performed on certain watermarking systems and is only relevant to certain watermarking applications. Many watermarking systems are designed to carry very small data payloads (of the order of 8 bits or fewer), with very high robustness, very low false positive probability and/or very low cost. Such systems typically employ codes that cannot be expanded to 80 bits. Furthermore, the tests performed by Stirmark are not critical to many applications that do not have stringent security requirements. They also do not represent a comprehensive test of the security required in applications in which an adversary is likely to have a watermark detector.

2.8.3 Scope of Testing

In general, watermarking systems should be tested on a large number of works drawn from a distribution similar to that expected in the application. Two important issues associated with these test criteria are the size and typicality of the test set which are especially important in testing the false positive rates. To truly verify this performance, it should be tested with

many millions of works even though in the real world it is very difficult to execute.

2.9 Summary

This chapter provides an overview of acoustic signals, its characteristics, human auditory system, different signal transforms and finally the watermarking concept. This section has been included for a quick reference to the concepts and definitions that are relevant to this research work.

Chapter 3

Review of Literature

3.1 Introduction

A better understanding in the field of audio watermarking is obtained by conducting a comprehensive literature review which leads to the identification of some of the unresolved issues within the existing implementations. The revised algorithms are diverse and so are grouped under different categories according to the methodology used in each algorithm. Different categories include time domain based algorithms, transform domain based algorithms and hybrid algorithms. This chapter also includes the evaluation techniques that are employed to identify the strength as well as the weakness of the proposed watermarking schemes.

3.2 Review of Audio Watermarking Algorithms

The watermarking algorithms suggested so far can be divided into different categories according to the methodology they employ in the embedding and

detection processes. Majority of these algorithms can be grouped into the following three categories:

- Time domain based algorithms
- Transformation based algorithms
- Hybrid algorithms

Following sections reveals the so far reviewed algorithms that belong to each of these three categories.

3.2.1 Time-Domain Based Algorithms

Time domain based algorithms embed the watermark in the time domain and therefore are easy to implement [Blackledge and Farooq 2008]. Many time domain based algorithms have been developed [Xiong et al. 2006; Ko, Nishimura, and Suzuki 2005; Oh et al. 2001; Liu and Smith 2004; Fujimoto, Iwaki, and Kiryu 2006] but these algorithms are less robust against attacks. A temporal audio watermarking algorithm based on cubic-spline interpolation scheme is described in [Deshpande and Prabhu 2009], which is tested with SQAM database. Different statistical techniques are often utilized to improve its robustness criteria as mentioned in the paper [Wang, Niu, and Yang 2009]. Least significant bit (LSB) based algorithms and echo hiding based algorithms are the two main algorithms reviewed in this category. An audio watermarking methodology based on spatial masking and general as well as synthetic ambisonics in [Nishimura 2012], presents the fact that a reversible watermarking is possible with the arrangement of loudspeakers in a playback fashion. The advantage of the scheme is that the first order ambisonics represents audio signals as mutually orthogonal four-channel signals thereby presenting a larger amount of data than in the original

version. Kais et al [Khaldi and Boudraa 2013] proposed a watermarking method in the time domain. The host signal is first segmented into frames and empirical mode decomposition (EMD) is conducted on every frame to extract the associated intrinsic mode functions (IMFs), then a binary data sequence consisting of synchronization codes and the watermark bits are embedded in the extrema of a set of consecutive last-IMFs. A bit (0 or 1) is inserted per extrema.

LSB Coding

LSB based watermarking is one of the primary techniques [Yeh and Kuo 1999; Cedric, Adi, and Mcloughlin 2000] in the field of multimedia watermarking including audio as well as other media types [Goljan, Fridrich, and Du 2001; Lee and Chen 2000; Fridrich, Goljan, and Du 1900]. The standard approach is to embed the watermark bits by altering the least significant bit values of selected samples in the digital audio. Detection process involves comparison of the altered values with the original values of samples.

The main advantage and disadvantage of this approach is that it can provide extremely high capacity and extremely low robustness because random changes of the signal can destroy the watermark [Mobasseri 1998]. It is very unlikely that the embedded watermark bits in those LSB regions can survive D/A and subsequent A/D conversions [Mobasseri 1998]. In addition, the modification of the sample values introduces a low power additive white gaussian noise (AWGN), which makes this algorithm less perceptually transparent because listeners are very sensitive to this noise [Cvejic and Seppanen 2005].

A major improvement on the standard LSB algorithm proposed in [Cvejic and Seppanen 2005] suggest that after embedding a watermark bit by

manipulating the i^{th} layer of the host audio using a novel LSB coding approach, alter its white noise properties by shaping the impulse noise occurred as part of the watermark embedding scheme. Experimental results shown that the imperceptibility achieved a MOS score of about 5.0, that is, it is perceptually transparent but this algorithm does not enhance the robustness to a significant extent.

Echo Hiding

Echo hiding based watermarking embeds a watermark bit by introducing an “echo” to the original signal. An echo represents the reflection of sound, arriving at the listener with a delay after the direct sound [Gruhl, Lu, and Bender 1996]. Four parameters of the echo used in the watermark embedding schemes are: initial amplitude, decay rate of the echo amplitude, “one” and “zero” offsets. As the offset between the original and the echo decreases the two signals blend. At a certain point, the human ear does not hear an original signal and an echo, but a single blended signal, depends on the quality of the original recording, the type of sound being echoed and the listener. The algorithm presented in [Gruhl, Lu, and Bender 1996] uses two different kernels, a “one” kernel which generates a “one” offset echo represented as a binary “1” and a “zero” kernel which generates a “zero” offset echo represented as binary “0”. Detection process first analyses the echoed signal cepstrum defined as the power spectrum of the logarithmic FFT power spectrum [Bogert, Healy, and Tukey 1963] which is followed by the autocorrelation to get the power of the signal [Gruhl, Lu, and Bender 1996]. Experimental results reveal that this algorithm is robust against MP3 compression, A/D and D/A attacks. The main disadvantage of this approach is that the detection rules are very lenient and therefore anyone can detect the watermark bits without requiring any key. Thus,

it is very vulnerable to malicious tampering [Ko, Nishimura, and Suzuki 2005; Katzenbeisser and Petitolas 2006].

Different echo hiding methods have been proposed to overcome the disadvantages of [Gruhl, Lu, and Bender 1996] algorithm. One method is the multi-echo embedding [Xu et al. 1999] but with limitations in the allocation of the delay time of the echoes. As an alternative to a single echo or multi-echo, [Ko, Nishimura, and Suzuki 2005] suggests a time spread (TS) echo where an echo is spread by a pseudo noise (PN) sequence [Ko, Nishimura, and Suzuki 2002] which results in its power spectrum being nearly flat in the mean-time sense and at the detection stage an additional de-spreading at the cepstrum has to be performed apart from the conventional approach. Thus, the use of PN Sequences enhances the security of the system which functions as the key in the detection stage but this algorithm is not sufficient enough to achieve a strong robustness. Different spreading kernels including chaotic sequences [Kubin 1995] and time-stretching pulses [Suzuki et al. 1995] are suggested to improve the robustness of the watermarking scheme. Conducting an ABX test to confirm the imperceptibility reveals that the proposed multiple-echo algorithm achieved a better imperceptibility than the single echo algorithm. A relatively detailed analysis on how each parameter affects the performance of the algorithm was provided in [Ko, Nishimura, and Suzuki 2005]. Another TS echo hiding algorithm was proposed in [Erfani and Siahpoush 2009] where the watermark bits were detected through the correlation between the cepstrum and the PN sequence. And the experimental results showed that it is less robust against signal processing attacks. The subjective listening test was acceptable with a MOS score of 4.7.

An improved, accurate and content based algorithm suggested in Erfani, Parviz, and Ghanbari 2007, for embedding the watermark bits iterates

the problems faced in the common echo hiding schemes by splitting the algorithm into two separate modules. The first module introduces a new time-spread algorithm which helps to embed the watermark bit as a whole and the second scheme helps to reduce or avoid the erroneously detected bits. In order to control the power of the watermark, fuzzy theory is employed in the embedding process which in turn provides a self-adaptive power to the embedded watermark in the original audio signal [Wang et al. 2008]. An enhanced time-spread echo hiding scheme presented in [Xiang et al. 2011] employs a new PN sequence which has frequency characteristics with smaller magnitudes in perceptually significant regions and has a correlation function with three times more large peaks than that of the existing PN sequence. In order to overcome the two security problems existing in the conventional time spread (TS) echo hiding schemes, the paper [Erfani and Siahpoush 2009] suggests two different schemes, the first scheme makes essential changes in the encoder and decoder of the TS echo hiding so that it reduces the problem of partly inserting the watermark bits into the audio signal. And the second scheme provides a content based TS echo hiding system based on the first proposed method that solved the second problem and reduces the problem of erroneously detecting the watermark bits even in the case of attack-free environment, of TS echo hiding.

Amplitude Masking

Auditory masking occurs when the perception of one sound is affected by the presence of another sound. Masking of one sound by another is depending upon various parameters such as frequency, amplitude, spectral or simultaneous masking. Amplitude masking in audio watermarking deals with the process of embedding the watermark bits into the host audio in

the form of an additional audio signal with very weak power, so that it can be masked by the effect of the original signal.

Gopalan et al. in [Gopalan and Wenndt 2004] take advantage of the idea on human hearing limitations and as shown in [Yang, Xingming, and Guang 2010] it embeds the watermark bits into a cover signal (human speech).

Phase Coding

Embedding watermark bits by altering the phase components of an audio signal can cause some problems because any sharp or radical alteration of phase from one frame to the other frame may result in audible phase inconsistencies. This scheme generally works by altering the phase of one piece of audio segment by the phase of another or simply by altering the phase of the original audio signal. Phase coding method is theoretically represented in the paper [Bender et al. 1996] which demonstrates that the success of phase coding technique is the result of the HAS to detect the phase changes within certain limits. In this scheme the cover audio is modified so that the phase embodies the information that need to be conveyed across. In the paper [Lipshitz, Pocock, and Vanderkooy 1982], the authors state that “Even quite small midrange phase nonlinearities can be audible on suitably chosen signals” which is supported by research in 2000 by [Koya 2000]. These schemes demonstrate that the phase changes from one frame to the next must be more gradual but sufficiently reduces its capacity. Work by [Yardimci, Cetin, and Ansari 1997] utilized all-pass filters with different phase characteristics to represent the embedded watermark bits. This may introduce some disturbances in the original signal which helps in discovering the presence of watermark in it. Employing dynamically varying phase characteristics towards watermark embedding is suggested in the work by [Takahashi, Nishimura, and Suzuki 2005].

3.2.2 Transformation Based Algorithms

Transformation based algorithms generally embed watermark bits by exploiting the desired properties of the data in the representation following the transformations that were carried out. Most popular transformations include the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and the Fast Fourier Transform (FFT) [Wang, Chen, and Chao 2004; Paquet and Ward 2002; Vieru et al. 2005; Quan and Zhang 2004; Wang and Zhao 2006; Ercelebi and Batakci 2009]. Some methods, such as Quantization Index Modulation (QIM), Singular Value Decomposition (SVD) and interpolation are often embedding the watermark bits like this. Watermark bits embedded in the transform domain of the signal are more robust against attacks and thus lead to the development of many different algorithms in this field.

DCT Based Watermarking Schemes

An HAS model is generally incorporated to minimize the perceptual distortion introduced in [Blackledge and Farooq 2008] but with less robustness. But incorporating HAS model increases the computation time [Kim et al. 2004], which restricts the use of these algorithms in time critical applications. Following are some of the typical algorithms in this category. The embedding algorithm [Tachibana et al. 2002] calculates a watermark signal in the frequency domain and converts it to the time-domain using an inverse DFT (IDFT). Then adds it into the host signal and the mis-synchronization attacks are handled by incorporating two-dimensional pseudo-random array (PRA), magnitude modification and non-linear sub-band. In a robust patch-work based watermarking scheme presented in [Natgunanathan et al.

2012], watermark bits are embedded into suitable DCT frame pairs to ensure high imperceptibility and the same criterion for the selection of DCT frame pairs are also applied at the decoding phase.

FFT Based Watermarking Schemes

Discrete Fourier Transform is a commonly used and powerful computational tool for performing the frequency analysis of discrete-time signals in the time domain and transforms this signal into its frequency domain [Jayshankar 2008]. A variety of algorithms have been proposed based on the components opted for embedding the watermark bits in the FFT spectrum. Most of the algorithms employ the magnitude of the FFT components [Megias, Herrera-Joancomarti, and Minguillon 2003; Megias, Herrera-Joancomarti, and Minguillon 2005; Fallahpour and Megias 2009; Megias, Serra-Ruiz, and Fallahpour 2010] and enhance the robustness criteria by incorporating a model of the HAS.

The scheme suggested in [Megias, Herrera-Joancomarti, and Minguillon 2003] chooses a set of frequencies by comparing the FFT spectrum of the original signal with that of the corresponding compressed-decompressed signal which disturbs the original signal at the most significant frequencies which is not desirable as far as perceptual transparency is concerned. Another scheme proposed in [Megias, Herrera-Joancomarti, and Minguillon 2005] introduces some randomness into the process of selecting the frequencies and improves its transparency at the price of losing some robustness. All these schemes come under the non-blind watermarking methodology which utilizes the spectrum of the original signal in the process of detecting the embedded watermark bits.

The algorithm proposed in [Fallahpour and Megias 2009] embeds watermark bits by manipulating the spline-interpolated magnitudes of the even

bins which are in turn derived from the spline-interpolation of the magnitudes of the odd bins. Towards its completion, the watermarked signal is reconstructed by inverse FFT. Both the embedding and detection of the watermark bits are performed on a frame-by-frame basis. This scheme offers a high capacity of 3000 bps and it is robust against most of the attacks defined in the StirMark Benchmark for Audio v1.0 [Xiang, Kim, and Huang 2008]. Since a magnitude comparison easily disturbs the suggested algorithm it would be vulnerable against certain attacks [Ercelebi and Batakci 2009] including the removal of the watermark bits.

In [Dhar, Khan, and Jong-Myon 2010], the watermark bits are embedded in the highest prominent magnitude of the spectrum by segmenting the original signal into non-overlapping frames. As in most of the cases, the extraction process is the exact inverse of the watermark embedding process. Even though this algorithm is very robust, the watermark information is easily removed since its embedding position is known. In [Megias, Serra-Ruiz, and Fallahpour 2010], the watermark bits are embedded by disturbing the magnitudes of the spectrum of the original signal at some selected frequencies by guaranteeing a good balance between imperceptibility and robustness. In this scheme, the robustness was evaluated based on a particular method defined in [Dittmann et al. 2006] than the conventional BER method which results in comparing it with other schemes.

An audio watermarking scheme suggested in [Khan, Xie, and Zhang 2010], utilizes the non-uniform DFT in embedding the watermark bits and its inverse function (IDTF) in the extraction process. This scheme employs fingerprints captured through a sensor as the labelled watermark. Another copyright protection scheme presented in [Fan and Wang 2009] employs discrete fractional Sine transform (DFRST) based watermarking scheme to improve the security of the system. This comes under a blind-

watermarking scheme and adopts normalized cross-correlation to confirm the similarity between the extracted and the original watermark. An FFT based watermarking algorithm proposed in [Viswanathan 2008] functions as a copyright protection scheme for ‘.wav’ files by embedding the copyright information such as the text messages into the frequency domain of the original files. This scheme could retain the perceptual quality of the audio signal and resistant to simple un-intentional and intentional attacks. In the work done by [Fan and Wang 2011], DFT coefficients with absolute distinct average values which are verified to be stable under playback speed modification are employed in the watermark embedding process. In the paper [Megias, Serra-Ruiz, and Fallahpour 2010], a novel time-domain synchronization technique is presented together with a new blind watermarking scheme which works in the DFT or FFT domain. The combined scheme provides excellent imperceptibility results while achieving robustness against typical attacks. A robust patch-work based watermarking algorithm for stereo audio signals are demonstrated in [Natgunanathan et al. 2013] by computing the DFT of the two sound channels and embedding the watermark bits to the selected DFT coefficients of each sub-segment pairs.

DWT Based Watermarking Schemes

The discrete wavelet transform (DWT) of a discrete-time signal is computed by applying successive low-pass and high-pass filtering which decomposes the signal into an approximation signal and a detail signal respectively. Both these signals are then downsampled by a factor of two [Filipe et al. 2007] and the process is carried out iteratively producing a set of approximation signals at different detail levels/scales and a final gross approxima-

tion of the signal. The maximum number of levels is determined by the length of the signal.

A standard DWT process is depicted in [Edwards 1991] where the original input is a 16 samples signal and ‘0’ means an actual system output. Utilizing the approximation coefficients in the watermark embedding is more robust compared to the detail coefficients; the reason behind this is that approximation coefficients contain more significant components [Keyvanpour and Merrikh-Bayat 2011] that can survive different attacks. As the level increases its robustness also increases but it reduces its capacity significantly and increases the computational cost to a great extent. A variety of DWT based watermarking algorithms have been proposed in [Fallahpour and Megias 2010; Al-Haj, Mohammad, and Bata 2011; Ercelebi and Batakci 2009; Bhat K, Sengupta, and Das 2010]. Watermark bits embedded into the DWT low frequency coefficients as presented in [Fallahpour and Megias 2010] which works by segmenting a long audio data sequence into many sections with a capacity of about 172 bps. But the embedding scheme does not give an idea on which level of coefficients are utilized in the watermark bits insertion scheme and could have an impact on the robustness and the capacity criteria. In [Al-Haj, Mohammad, and Bata 2011], the watermark is embedded in the second level of ‘detail’ coefficients to achieve an acceptable trade-off between robustness and imperceptibility and the author suggests that the two-level DWT gives a better result than a higher level DWT with a non-blind detection scheme. Employing the first level DWT approximation coefficients in the watermark embedding scheme with the help of pseudorandom sequence to improve the security of the algorithm is suggested in [Ercelebi and Batakci 2009]. Performance is calculated with an SNR of approximately 30 dB for all the test samples and MOS score of 5.

The first adaptive DWT-SVD audio watermarking scheme is proposed in [Bhat K, Sengupta, and Das 2010] which is influenced by an image watermarking scheme presented in [Fridrich, Goljan, and Du 2001]. In [Bhat K, Sengupta, and Das 2010], the embedding process derives the first level DWT approximation coefficients for each frame and then organized as a two dimensional data matrix which is then decomposed using the SVD and the coefficients in one of the decomposed matrices are altered [Dumitrescu, Wu, and Wang 2003], by which the watermark bits are embedded. Experiments demonstrated that this [Bhat K, Sengupta, and Das 2010] algorithm achieved a good imperceptibility with an SNR of about 24.37 dB and a MOS score of 4.46 and is robust against attacks such as MP3 compression, lowpass filtering and additive noise with capacity of about 45.9 bps. An algorithm designed to cope with D/A and A/D conversions is presented in the work done by [Xiang 2011], which inserts the watermark in the DWT domain using the relative relationships among different groups of the DWT coefficients.

Based on wavelet moments and synchronization codes [Wang, Niu, and Lu 2011], demonstrates an audio watermarking scheme which inserts the watermark bits into the average value of modulus of the low-order wavelet moments obtained for each audio segment. Experimental results showed that the suggested scheme is robust against common signal processing and de-synchronization attacks by keeping its audio qualities. A robust and blind watermarking scheme for medical breathe sounds by combining lifting wavelet transform (LWT), discrete cosine transform (DCT), singular value decomposition (SVD) and dither modulation (DM) quantization are suggested in [Lei, Song, and Rahman 2013].

In SVD scheme presented in [Lei, Soon, and Li 2011], the embedding scheme is described as follows: Step 1 - Select the coefficient pairs to be

modified in the potential positions of the SVD-DCT block; Step 2 - Compute the frequency mask; Step 3 - Use the frequency mask to weight the watermark amplitude; Step 4 - Modify the relationship between the selected coefficient pairs according to the watermark embedding rule. Extraction is done by summing the value of the difference between the coefficients of each selected pair. Employing wavelet based entropy scheme towards the watermark embedding is presented in [Chen et al. 2013]. This arranges the watermark bits after converting synchronization codes into a binary pseudo-random noise (PN) sequence, $B = \beta_l$, which is embedded into the lowest-frequency coefficients of its DWT. And the synchronization codes are used to locate the watermark bits in the detection process.

Spread Spectrum Based Watermarking Schemes

Spread-Spectrum (SS) communications pave the way to the development of Spread-Spectrum watermarking schemes. In SS communications, a narrowband signal is modified so that its energy is spread over a much larger bandwidth and as a result, the signal energy present in any single frequency is almost undetectable. Similarly, in SS watermarking, the watermark energy is spread over many frequency bins so that the energy in any one bin is very small and is difficult to detect [Cox et al. 1997].

Generally an SS based watermarking algorithm can be described as follows [Kirovski and Malvar 2003]: Let the data to be watermarked be ‘y’ which represents a collection of samples obtained by performing an appropriate invertible transformation on the original audio signal and the watermark to be embedded is defined as a direct SS sequence ‘w’, a pseudo-randomly generated vector mutually independent of ‘y’. Embedding is done as per the equation $y' = y + \delta w$ and detection is performed by correlating

y' with w and the energy of watermark is very low as it is distributed across all the frequency bins.

Various algorithms have been proposed in the field of SS watermarking methodology which includes [Cox et al. 1997; Kirovski and Malvar 2003; Cox et al. 2002; Wolfgang, Podilchuk, and Delp 1999; Swanson et al. 1998; Wu, Su, and Kuo 1999; Malvar and Florencio 2003]. In [Cox et al. 1997], robustness of the watermarking is maximized by employing the perceptually significant components of the signal spectrum for the embedding process as well as it exploits the perceptual masking phenomena of the signal without degrading the imperceptibility. An oversampled filter bank that provides a perfect reconstruction of the signal termed as the modulated complex lapped transform (MCLT) [Malvar 1999] is employed in the watermarking scheme suggested in [Kirovski and Malvar 2003]. Robustness of this scheme is evaluated by the attacks defined in Stirmark audio [Steinebach, Lang, and Dittmann 2002] and all but one attack had a minimal effect on the correlation value. Performance of the algorithm can be improved by incorporating some block repetition coding so as to protect against desynchronization attacks [Anderson and Petitcolas 1998], cepstrum filtering, improved robustness and audible MCLT sub-band detection to improve the imperceptibility. All these processes help to overcome the trade-off between imperceptibility and robustness to a great extent.

A frequency selection based SS watermarking scheme presented in [Malik, Ansari, and Khokhar 2008] uses only a fraction of the audible frequency range for embedding the watermark bits and employs a secret key to select sub-bands for embedding. The suggested algorithm is robust against low-pass filtering but the capacity and imperceptibility were not evaluated. However, this algorithm is computationally intensive by incorporating various processes including the HAS model [Brian Loker 2002]. Select-

ing pseudo-Zernike moments which tolerates common signal processing and de-synchronization attacks for embedding the watermark bits presented in [Wang, Ma, and Niu 2011] is treated as robust, blind watermarking scheme with good audio quality. A robust and low complexity audio watermarking scheme [Ercelebi and Subasi 2005], which embeds multiple bits of the watermark in each audio sub-frame enhances the robustness and taper-resistance of the embedded watermark and thereby provides high recovery rate compared to common watermarking algorithms introduced so far. [Chen and Xiao 2013] proposes a new perceptual audio hashing algorithm to enable content-oriented search in database. The system runs through the following steps: framing, mapping each frame to 2-D form, Zernike transforming, generating virtual watermarks and finally the maximum likelihood (ML) watermark detection and hash generation. And the suggested scheme performs satisfactorily in the discrimination, perceptual robustness and identification rate.

In the work [Kang, Yang, and Huang 2011], the authors introduced a multi-bit spread-spectrum audio watermarking scheme based on a geometric invariant log coordinate mapping (LCM) feature. In this scheme, the watermark bits actually embedded in the DFT domain via LCM could completely eliminate the distortions resulting from the non-uniform interpolation mapping. An additive spread spectrum watermarking scheme proposed in [Wook Kim et al. 2010] incorporates a selective correlation detector (SCD) to utilize a portion rather than the entire host signal towards the embedding process.

3.2.3 Hybrid Algorithms

Hybrid algorithms refer to those algorithms that differ from the above two schemes due to its novelty. Main algorithms in this scheme include i) based

on Chirp Coding [Blackledge 2007], ii) patchwork [Kalantari et al. 2009] and iii) based on the SVD [Xiang, Kim, and Huang 2008] which are explored in the following sections. A semi-fragile audio watermarking scheme presented in [Wang and Fan 2010] generates its watermark by taking the hash value of the centroid of each audio frame and inserts into the hybrid domain by taking its DWT and DCT transformations.

Chirp Coding Based Watermarking Schemes

A fragile watermarking algorithm proposed in [Blackledge 2007] performs 7 levels of wavelet decomposition of the signal to produce 7 levels of detail coefficients. Detail coefficients are very sensitive to attacks such as lossy compression and audio slicing and that is the reason for using these coefficients in the embedding scheme. Upon preparation of the watermark bits, a chirp function is created which is then multiplied with a new signal created based on the binary sequence and scaled by a predefined scale factor to produce the chirp code. This chirp code with a very low frequency and amplitude is then added to the original signal to generate the watermarked signal. Detection of the watermark is performed by applying the same chirp function and a correlation function on the watermarked signal.

Quantization Based Watermarking Schemes

Quantization based watermarking scheme is one of the simplest blind watermarking schemes introduced so far. [Chen and Wornell 2001] defines a QIM function as given in the following equation 3.1.

$$s(y, i) = q_{\Delta}(y + d[i]) - d[i] \quad (3.1)$$

where $q_{\Delta}(\cdot)$ denotes a uniform scalar quantizer with step size Δ , 'y' is the vector to be quantized, 'i' represents the watermark bit number and 'd[i]' is the dither vector with 'N' number watermark bits. Dither vector can be defined as follows:

$$d[i, 1] = \begin{cases} d[i, 0] + \frac{\Delta}{2} & , d[i, 0] < 0, \\ d[i, 0] - \frac{\Delta}{2} & , d[i, 0] \geq 0. \end{cases} \quad (3.2)$$

where $i = 1, 2, N$; $d[i, 1]$ corresponds to the vectors for embedding a watermark bit '1' and $d[i, 0]$ corresponds to a watermark bit '0'. Generally, quantization is applied on the coefficients obtained as a result of the FFT or DWT transformation on the signals [Kalantari, Ahadi, and Kashi 2007; Singh, Garg, and De 2012].

The watermark embedding process proposed in [Kalantari, Ahadi, and Kashi 2007] works as follows: at first the original signal is DWT transformed, then based on the mean value of the DWT coefficients the step size Δ is calculated which is followed by the dither vector calculation. Modulating the mean value of the DWT coefficients returns new coefficients that can be used in the embedding process. Finally, the inverse DWT is applied to reconstruct the time domain watermarked signal. But the suggested algorithm cannot survive filtering and echo attacks but has a capacity of about 86 bps. Another scheme [He and Scordilis 2006] suggested that decompose the signal into critical band-like partition and select the wavelets so as to meet the required temporal resolution. According to this scheme, the derived masking threshold is $T = \max(T^s(i), T^p(i))$, where $T^p(i)$ is the temporal masking thresholds for the current frame and $T^s(i)$ is the frequency masking threshold derived by the proposed DWPT.

A time-spread watermarking scheme in [Xiang et al. 2012] divides the host-signal into two sub-signals and embeds the watermark bits by utiliz-

ing the cepstral coefficients, the segments are combined to make the watermarked signal and send across the channel. At the detection phase, similarity of the cepstra corresponding to the watermarked sub-signals is exploited to extract the embedded watermarks. A hybrid LWT-SVD audio watermarking method introduced in [Lei et al. 2012] efficiently inserted in the coefficients of the LWT low frequency sub-band taking advantage of both SVD and quantization index modulation (QIM), results in a semi-blind watermarking scheme. And towards the end of the embedding process, inverse LWT is conducted to reconstruct the watermarked signal.

Patchwork Algorithms

The patchwork algorithm was first used for image watermarking by Bender [Bender et al. 1996]. The idea behind this approach is to select two data sets or patches randomly from the host signal and the mean values of these two patches are manipulated by a constant value, which in turn defines the watermark strength. The detection process starts with the subtraction of the sample values between the two patches, which decides the presence of the watermark without using the host signal. Even though the patchwork algorithms itself are very good, they work on an assumption that the expected value of the differences of the two sample means should always be ‘0’ but in practice the actual difference between these sample means is not always ‘0’ [Kalantari et al. 2009; Yeo and Kim 2003b].

An audio patchwork watermarking scheme was first proposed by M Arnold in [Arnold 2000]. Several works have been conducted as an improvement on this scheme. Modified patchwork algorithm (MPA) presented in [Yeo and Kim 2003b] and the generalized patchwork algorithm (GPA) proposed in [Yeo and Kim 2003a] are some examples. MPA works by selecting four patches instead of the two patches that originally used, with two of

these four patches are used to embed the watermark bit ‘1’ and the other two are used for the watermark bit ‘0’. GPA was proposed as a generalization to the MPA method by employing both additive and multiplicative embedding rules.

The work presented in [Kalantari et al. 2009] deals with a multiplicative patchwork method (MPM) for audio watermarking. Watermark bits are embedded in the wavelet domain of the host signal by multiplying or dividing the samples of one set out of the two data sets selected and leaves the other unchanged. Perceptual quality of the watermarked audio is calculated for each iteration by incorporating the PEAQ algorithm and this in turn determines the watermark strength to get a better imperceptibility. But this algorithm is vulnerable to phase changes and its computational efficiency is very low due to the incorporation of PEAQ for each iteration. The host-signal is segmented into non-overlapping frames with the length of N and then the watermark data are simply embedded by scaling the amplitude of each frame. Decoder extracts the watermark data using the host signal variance, noise variance and watermark strength factor for each frame. Thus the decoder acts in a semi-blind way [Akhaee, Kalantari, and Marvasti 2010].

Interpolation Based Watermarking Schemes

Interpolation denotes the technique of constructing new data points within the range of a specific set of discrete data [Fallahpour and Megias 2009; IEEE 1979; Fujimoto, Iwaki, and Kiryu 2006]. Polynomial interpolation presented in [Fallahpour and Megias 2009] is one of the best known interpolation approaches which is simple to implement and a good quality of interpolant can be obtained from it. Another scheme suggested in [Deshpande and Prabhu 2009], demonstrates a spline interpolation based wa-

termarking scheme. In the embedding process, the time domain signal is divided into frames and each frame is further divided into four groups called $\zeta_1, \varphi_1, \varphi_0, \zeta_0$ and for each watermark bit ‘0’, the samples in φ_0 are replaced by the values interpolated from ζ_0 , while the samples in φ_1 are unchanged; in the other case, ie. for each watermark bit ‘1’, the samples in φ_1 are replaced by the values interpolated from ζ_1 , while the samples in φ_0 are unchanged. Similarly, in either case, the samples in ζ_1 and ζ_0 are left unchanged. The watermarked signal is then reconstructed. Detection process is the reverse of the embedding scheme with a calculation of ρ_1^2 and ρ_0^2 based on the mean squared error.

Earlier works [Fujimoto, Iwaki, and Kiryu 2006; Martin, Chabert, and Lacaze 2008] suggested the replacement of all the samples in set φ_1 and φ_0 with the interpolated values if the watermark bit is a ‘1’ and left unmodified if the watermark bit is a ‘0’. Imperceptibility of the scheme presented in [Deshpande and Prabhu 2009] is assessed by conducting an ABX subjective listening test which reveals that perceptual distortion increases with increase in frame size.

SVD Based Watermarking Schemes

SVD is a well-known numerical analysis tool used on matrices to compute its singular values. Let A be an arbitrary $m \times n$ matrix, with the full SVD, then it can be decomposed as 3.3:

$$A = USV^T \quad (3.3)$$

where U represents an $m \times m$ unitary matrix and V corresponds to an $n \times n$ unitary matrix, called the left eigenvector and the right eigenvector respectively, both U and V are orthogonal. S is an $m \times n$ diagonal matrix

with singular valued elements in it, which are the square root of the eigenvalues [Liu, Niu, and Kong 2006]. Now, SVD has been used extensively as an effective tool in digital watermarking [Bhat K, Sengupta, and Das 2010; Liu, Niu, and Kong 2006; Ali and Ahmad 2010; Chung et al. 2007; Fan, Wang, and Li 2008; Chang, Tsai, and Lin 2005; Ganic, Zubair, and Eskicioglu 2003; Chandra 2002; Sun, Sun, and Yao 2002; Ozer, Sankur, and Memon 2005; Zhang and Li 2005; Kong et al. 2006; Emek and Pazarci 2006; Hu and Chen 2007; Trefethen and Bau III 1997; Mohammad, Alhaj, and Shaltaf 2008; Liu and Tan 2002]. Most of the existing SVD based watermarking algorithms have been applied to images [Liu, Niu, and Kong 2006; Fan, Wang, and Li 2008; Chang, Tsai, and Lin 2005; Ganic, Zubair, and Eskicioglu 2003; Chandra 2002; Sun, Sun, and Yao 2002; Ozer, Sankur, and Memon 2005; Zhang and Li 2005; Kong et al. 2006; Emek and Pazarci 2006; Hu and Chen 2007; Trefethen and Bau III 1997; Mohammad, Alhaj, and Shaltaf 2008; Liu and Tan 2002; Chung et al. 2007]. SVD based audio watermarking schemes exist but are limited [Bhat K, Sengupta, and Das 2010; Ali and Ahmad 2010]. SVD based watermarking algorithms can be mainly grouped into two categories. The first category of these algorithms is ‘informed’ or ‘non-blind’ [Zhang and Li 2005; Mohammad, Alhaj, and Shaltaf 2008; Liu and Tan 2002], requiring access to the original signal or the watermark towards the successful detection of the embedded watermark. The second category does not require the original signal for detecting the embedded watermark and hence comes under the ‘blind’ watermarking scheme. These schemes are based on outcome of the work suggested in [Bhat K, Sengupta, and Das 2010; Ali and Ahmad 2010; Chung et al. 2007; Fan, Wang, and Li 2008; Chang, Tsai, and Lin 2005]. In [Chung et al. 2007; Fan, Wang, and Li 2008; Chang, Tsai, and Lin 2005], image watermarking algorithms using the SVD tools are demonstrated. Audio

watermarking schemes presented in [Bhat K, Sengupta, and Das 2010; Ali and Ahmad 2010] shows that changing ‘S’ to some extent does not affect the audio quality significantly and that singular values in ‘S’ are consistent after common signal manipulations. However, there are some drawbacks in manipulating ‘S’ as discussed in [Bhat K, Sengupta, and Das 2010; Ali and Ahmad 2010]. First one is that, the modification of the largest coefficients in ‘S’ makes the embedding position noticeable [Andrews and Patterson III 1976]. Second one shows that, the modification of the less significant coefficients in ‘S’ results in distorting the signal after signal processing operations [Andrews and Patterson III 1976; Ranade, Mahabalarao, and Kale 2007]. Finally, the number of singular elements in ‘S’ which are available for manipulation is very limited resulting in a scheme with sufficiently low capacity [Chang, Tsai, and Lin 2005].

As far as the performance of the SVD based audio watermarking algorithm is concerned, a non-blind algorithm proposed in [Ozer, Sankur, and Memon 2005] embeds the watermark bits by manipulating the coefficients in the matrix ‘S’, to generate a new matrix S_w . Then, decomposition is carried out on S_w by the SVD as U_w, S'_w, V_w . Another non-blind audio watermarking algorithm suggested in [Ali and Ahmad 2010] can be used to illustrate the performance of the SVD based audio watermarking algorithm. The work presented in [Krishna Kumar and Sreenivas 2007] suggested the watermark-to-host correlation (WHC) of random phase watermarks, in the context of additive embedding and blind correlation-detection of the watermark in audio signals. And it is concluded that WHC is higher when a legitimate watermark is present in the audio signal. The algorithm implemented in [Al-Nuaimy et al. 2011] proceeds as follows: The 1-D audio signal is transformed into a 2-D matrix (A matrix), the SVD is performed on the A matrix, the chaotic encrypted watermark (W matrix) is added to

the SVs of the original matrix, the SVD is performed on the new modified matrix (D matrix), the watermarked signal in 2-D format (A_w matrix) is obtained using the modified matrix of SVs (S_w matrix) and finally the 2-D A_w matrix is transformed again into a 1-D audio signal. Reverse process of the embedding scheme reveals the presence of watermark in the signal.

Another SVD based watermarking scheme presented in [Bhat K, Sen-gupta, and Das 2010] embeds the watermark bits by applying a QIM process on the singular values in the SVD of the wavelet domain blocks. The work suggested in [Wang, Niu, and Lu 2011] has a threshold based distortion control technique which is applied to preserve the audio fidelity characteristics; its embedding scheme is based on the reduced singular value decomposition (RSVD).

Muteness Based Watermarking Schemes

A watermarking scheme introduced by [Kaabneh and Youssef 2001] introduces the replacement of muteness for embedding the watermark bits. This is achieved by first inserting the watermark bits '0' and '1' without changing the mute period, then increase the mute period slightly by a value for a '0' and a different value for a '1'. This technique is a time domain embedding technique and the periods of relative quiet or silence can be utilized to embed the watermark bits. But a disadvantage of this scheme also occurs; the transmission through a noisy channel may introduce noise in the area of muteness that may result in the degradation of the watermark detection process. An improvement on this scheme is suggested in the work done by [Wu et al. 2011] by adjusting the length of the mute period dynamically.

IPR Related Watermarking Algorithms

The paper [Faundez-Zanuy, Hagemuller, and Kubin 2007] demonstrates a security enhanced biometric incorporated speaker identification system, which can be employed in forensic applications. This scheme embodies a time-stamp to keep a track on whether the speech signal is up to date or has already expired and functions by employing a combination of two security mechanisms such as biometrics and watermarking. In order to prevent unauthorized copying of digital audio, a psychoacoustic model is designed which embeds the watermark into the frequency domain of the signal by employing the DSSS methodology [Seok and Hong 2001]. Another scheme suggested in [Seok, Hong, and Kim 2002] also inserts the watermark bits by DSSS methodology and employing the LPC parameters is also a copyright protection scheme which includes a psychoacoustic model of audio coding. A watermarking scheme which resolves rightful ownership as well as to authenticate the customer's legitimacy in images is presented in the work [Lin 2001]. A content-based audio authentication scheme introduced in [Gulbis, Muller, and Steinebach 2009] embeds the watermark data in the same domain as the content feature extraction without causing any impairment of the same. Automatic identification of audio recordings of ethnic music by employing hidden Markov models (HMMs) is presented in [Orio 2010]. Research on psychoacoustics exploits the use of embedding an arbitrary message, the watermark, in a digital recording without altering sound perception [Boney, Tewfik, and Hamdy 1996]. In commercial systems the watermark can be any data which reveals the identification of the item/recording (such as title, author, performers, the copyright owner and even the user that purchased the digital item) [*The Impact of the Internet on Intellectual Property Law*]. A cyclic pattern embedding watermarking algorithm to detect speech forgery is presented in the paper [Park, Thapa,

and Wang 2007]. An audio dependent watermarking procedure suggested in the work [Swanson et al. 1998] creates its watermark by segmenting the audio clips and adding a perceptually shaped pseudo-random sequence, which in the complete system acts as the author signature and guarantees the copyrighting of each time-domain signal. The work implemented in [Yucel and Ozguler 2010] helps to identify the owner and the distributor of digital data by employing the frequency domain of each multimedia files.

An author identification scheme with speech watermarking [Faundez-Zanuy, Hagemuller, and Kubin 2006] utilizes a combination of spread-spectrum methodology with frequency masking, inserts a speaker authentic bio-metric signal with an expiration based time-stamp as the watermark. Employing counter-propagation neural networks (CPN) in the context of audio watermarking is presented in [Chang, Wang, and Shen 2010] to confirm the copyright authentication and the synchronization codes are embedded in the low-frequency coefficients of the candidate frame. In the work presented in [Lei and Soon 2012], a new audio watermarking technique for content integrity authentication and copy-right protection by combining chaotic compound encryption and synchronization techniques is proposed. This modifies the most significant DCT coefficients of the corresponding scrambled positions in the host audio for copyright protection and alter insignificant DCT coefficients for content authentication. A procedure for logo watermarking of speech signal presented in [Orovic et al. 2008] selects the DFT coefficients on time-frequency basis and inserts the logo in it. And the speech regions suitable for the watermark are selected by using the S-method.

[Dutta, Gupta, and Pathak 2012] presents a perceptible watermarking (P - watermarking) for encrypting the audible data into inaudible data as the first step and imperceptible watermarking (I-watermarking) for taking

care of the copyright issues. A signal-dependent sequence using modified Swanson's method is suggested [Tsai and Cheng 2005] as the watermark and ANN is used towards embedding the watermark. An authentication scheme for music audio is introduced in [Yoshitomi et al. 2011] using DWT in which the audio is authenticated using the features extracted by the wavelet transform and characteristics coding, and provides 100% authentication ratio. In the work done by [Steinebach and Dittmann 2003], two concepts for digital audio content authentication were introduced. First one is the content-fragile watermarking which is based on combining robust watermarking and fragile content features. Second one is the invertible watermarking approach which is applicable for high-security scenarios. In order to overcome the problem of backward-compatible audio authentication, a distributed source coding based scheme presented in [Varodayan, Lin, and Girod 2008] is robust against legitimate encoding variations and detects any illegitimate modifications.

Other Watermarking Algorithms

A robust audio watermarking scheme against desynchronization attack is proposed by [Wang, Niu, and Qi 2008] which employs audio characteristics and synchronization codes. This scheme works by utilizing the SVM theory towards the identification of optimal embedding positions, in the next step, employs the 16-bit Barker code as the synchronization mark and finally the watermark bits are embedded into the statistical average value of low-frequency components in wavelet domain by making full use of auditory masking. In the work presented in [Hussain 2013], S-box transformation is used in the embedding process; the system can be described through the following steps: audio file \rightarrow transform into matrix \rightarrow convert each value of

pixel into binary form having eight bits \rightarrow S-box transformation \rightarrow transform modified matrix into audio file \rightarrow play audio \rightarrow end. A good fidelity learning-based audio watermarking scheme using kernel Fisher discriminant analysis (KFDA) is presented in [Peng et al. 2013] which embodies two techniques in the watermarking module i.e., down-sampling technique and energy relationship modulation technique. Another scheme proposed in [Yang, Wang, and Ma 2011] denoises the signal and then divides each segment into two parts. Synchronization code is then embedded into the statistical average value of audio samples and watermark bits are embedded in the wavelet domain using the higher-order statistics, obtained by using the Hausdorff distance. A statistical mean manipulation method utilized in the watermark embedding scheme presented in the work by [Hu and Chen 2012] is based on a supplementary cepstrum-based scheme.

An algorithm suggested in [Li et al. 2011] embeds the pre-processed 1-dimensional binary audio watermarking in the statistical average of the sampling value of pre-processed host audio signals. Watermark extraction is based on the knowledge about the maximum likelihood decoding. A watermarking scheme based on back-propagation neural network (BPNN) architecture which hides the watermark bits into the middle frequency-band after performing a DCT is presented in [Charfeddine, El" arbi, and Amar 2012]. A set of content-based audio watermarking schemes presented in [Xu and Feng 2002] works for WAV audio, WAV-table synthesis audio and compressed audio is based on the audio content as well as on the HAS.

3.3 Evaluation Strategy of Watermarking Schemes

Important factors in the evaluation of watermarking algorithms in practical applications rely on the following criteria and thus the advantages and

disadvantages of each of these algorithms are defined accordingly:

- The performance of the algorithm in terms of imperceptibility, robustness, capacity, computational efficiency and blindness characteristics
- The reliability of the results obtained for each algorithm
- Establishing the hardness in extracting the embedded watermark bits
- Establishing whether the algorithms incorporate any additional processes to bypass the trade-off between imperceptibility and robustness criteria
- Establishing the effect of parameters in the performance of the algorithm by a detailed analysis of the same.

The above mentioned criteria are useful for determining whether the algorithms under review are worthy of further research.

3.3.1 A Quantitative Approach to the Performance Evaluation

Performance of the watermarking algorithms are evaluated for imperceptibility, robustness, capacity and computational efficiency since these are the important characteristics of every watermarking schemes.

Imperceptibility

In general, perceptual quality of audio is evaluated by [Bhat K, Sengupta, and Das 2010]:

- Subjective evaluation by human listening tests

- Objective evaluation by signal-oriented measures such as the signal to noise ratio (SNR)

In order to conduct the subjective evaluation tests an ABX test [Deshpande and Prabhu 2009] has been employed where the objective listener is provided with three audio files, let it be X (original audio file), X' (water-marked audio file) and Y' (can be either X or X'). The goal of the listener is to identify whether Y' is X or X' . Another scheme [Lei, Soon, and Li 2011] employs, the mean opinion score (MOS) [ITU-T 1996] to evaluate the imperceptibility criteria. But as denoted by [Arnold and Schilz 2002], employing human listening tests towards subjective evaluation takes more time and the results varies with different listeners.

In order to overcome the drawbacks of subjective quality evaluation, an objective measurement such as the SNR is widely agreed. It can be evaluated using the following equation 3.4.

$$SNR = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n [x(n) - x'(n)]^2} \quad (3.4)$$

where $x(n)$ denotes the original audio signal and $x'(n)$ denotes the water-marked signal.

Accuracy

The accuracy of a watermarking algorithm is defined as the precision of detecting the embedded watermark without undergoing any attack or loss of it. It can be measured by the bit error rate (BER) [Bhat K, Sengupta,

and Das 2010], which is defined as follows:

$$BER(W_1, W_2) = \frac{\sum_{i=1}^N W_1(i) \oplus W_2(i)}{N} \quad (3.5)$$

where W_1, W_2 corresponds to the original and the detected watermark bit sequences respectively, 'N' denotes the total number of bits and 'i' denotes the respective bit number. In this work, the performance of the accuracy is evaluated by Precision which is more straight forward and is defined as follows:

$$Precision(W_1, W_2) = N - \frac{\sum_{i=1}^N W_1(i) \oplus W_2(i)}{N} \quad (3.6)$$

The denotation of each variable is the same as that defined in Equation 3.5. If 'N' audio signals are used in an experiment, the mean precision denoted as $Precision_{mean}$, is calculated as follows with 'i' denotes the signal number.

$$Precision_{mean} = \frac{\sum_{i=1}^N Precision_i}{N} \quad (3.7)$$

Robustness

The robustness of a watermarking algorithm is defined as its accuracy in detecting the watermark after experiencing different attacks during the signal transmission. It also can be evaluated by the BER [Bhat K, Sengupta, and Das 2010] measure. Likewise, Precision can also be used to evaluate the robustness.

Generally, in order to improve the robustness of the watermark embedding scheme, the same watermark bit sequences are embedded repetitively

and employing the ‘mode’ operation at the detection side helps in identifying the data that occurs most frequently in a defined data set [Griffiths 2008].

Capacity

The capacity of the watermarking channel is defined as the maximum number of bits that can be embedded into the selected cover signal and expressed as the number of bits per second (bps). Suppose the length of the host audio is L seconds and the number of embedded watermark bits is M [Bhat K, Sengupta, and Das 2010], then the capacity is M/L bps.

Computational efficiency

The computational efficiency purely depends on the implementation platform and can be measured as the CPU time required for the watermark embedding and detection processes.

Some of the milestones in the field of audio watermarking can be understood from the survey of [Spanias, Painter, and Atti 2006].

Other characteristics (adapted from [Wang 2011]) than those presented in tables - table 3.1, table 3.2 and table 3.3 are described below:

- **Blindness:** refers to whether or not the original signal and the watermark is needed to detect the presence of the embedded watermark.
- **Additional processes:** refers to those processes that are incorporated to improve the performance of the algorithm.
- **Parameters analysis:** refers to whether or not an analysis is carried out to determine the parameters that affect the performance.

- **Removability:** refers to whether the watermark is easy to detect/remove or not.
- **Reliability:** refers to whether the performance given could be said to be reliable or not which really depends on whether the test data was sufficiently large, and was taken across a variety of genres.

Below tables - table 3.1, table 3.2 and table 3.3, summarize the four main performance characteristics of the typical audio watermarking algorithms that were reviewed.

Table 3.1: Performance comparison

Algorithm	Imperceptibility	Robustness	Capacity	Efficiency
LSB [Cvejic and Seppanen 2005]	5.0	low	44100	high
Echo hiding [Ko, Nishimura, and Suzuki 2005]	5.0	low	n/a	n/a
FFT [Fallahpour and Megias 2009]	4.5	high	3000	high
DWT [Bhat K, Sengupta, and Das 2010]	4.6	high	47	low
SS [Kirovski and Malvar 2003]	n/a	high	n/a	low
QIM [Kalantari, Ahadi, and Kashi 2007]	n/a	average	86	n/a
Patchwork [Kalantari et al. 2009]	4.4	high	13	low

Table 3.2: Performance comparison (contd..)

Algorithm	Imperceptibility	Robustness	Capacity	Efficiency
Interpolation [Deshpande and Prabhhu 2009]	5.0	average	3000	n/a
SVD [Özer, Sankur, and Memon 2005]	4.7	high	n/a	n/a
Spatial Masking [Nishimura 2012]	n/a	average	n/a	n/a
EMD [Khaldi and Boudraa 2013]	good	good	n/a	n/a
Echo [Erfani, Parviz, and Ghanbari 2007]	4.5%	good	n/a	n/a
Echo [Wang et al. 2008]	good	better	n/a	n/a
Echo [Xiang et al. 2011]	improved	improved	n/a	n/a
Echo [Erfani and Siahpoush 2009]	4.7%	good	n/a	n/a
Zernike [Wang, Ma, and Niu 2011]	Good	Good	n/a	n/a
Zernike [Chen and Xiao 2013]	Good	Good	n/a	n/a
PRS [Ercelebi and Subasi 2005]	Good	Good	1023 bits	n/a
DWT-x/y [Chen et al. 2013]	4.4/4.9%	Strong	2000/1000 bits	n/a
Zernike [Wang, Ma, and Niu 2011]	Good	Good	n/a	n/a
Zernike [Chen and Xiao 2013]	Good	Good	n/a	n/a

Table 3.3: Performance comparison (contd..)

Algorithm	Imperceptibility	Robustness	Capacity	Efficiency
PRS [Ercelebi and Subasi 2005]	Good	Good	1023 bits	n/a
DWT-x/y [Chen et al. 2013]	4.4/4.9%	Strong	2000/1000 bits	n/a
Zernike [Chen and Xiao 2013]	Good	Good	n/a	n/a
PRS [Ercelebi and Subasi 2005]	Good	Good	1023 bits	n/a
DWT-x/y [Chen et al. 2013]	4.4/4.9%	Strong	2000/1000 bits	n/a

Tables - table 3.1, table 3.2 and table 3.3, reveal that different algorithms have different strengths and weaknesses. It can be summarized that the FFT based watermarking algorithm proposed in [Fallahpour and Megias 2009], the DWT based watermarking algorithm proposed in [Bhat K, Sengupta, and Das 2010], and the SVD based watermarking algorithm proposed in [Ozer, Sankur, and Memon 2005] achieved a better overall performance compared to the others. The computational efficiency of the algorithm proposed in [Fallahpour and Megias 2009] is very high which makes it very attractive for time-critical applications, and it has an extremely high capacity. But, the imperceptibility of [Fallahpour and Megias 2009] is not too satisfactory. Another issue with [Fallahpour and Megias 2009] is that its embedding scheme is based on some magnitude comparisons which makes it vulnerable to attacks that result in magnitude distortions. The elements selected for embedding the watermark bits are intrinsically robust against signal processing attacks and it makes the SVD a powerful tool for audio watermarking.

Table 3.4: Advantages and disadvantages of watermarking schemes (contd..)

Technique	Advantages	Disadvantages
Least significant bit	Computationally simple, Mature technique in image-processing, Resistant to time-shifting of the audio, Can be decoded without original or watermark.	Least-significant bit is most likely to be altered by typical signal manipulations, Not suitable to audio processing due to sensitivity of auditory system, Masking techniques would be inconsistent.
Echo-hiding	Human auditory system often cannot perceive very short echo, Audible added echo can cause sound to be perceived as 'warmer', Addition of 'pre-echo' can improve effectiveness	Weak against time-domain compression and time-shifting, Low-amplitude echo susceptible to interference from channel noise, original or watermark required for decoding.
Amplitude masking	Computationally simple, Ineffective in signals with low-powered components	Not suitable for real music, Problems encountered with interference from inherent components.
Phase coding	Human auditory system generally unable to detect phase	Phase inconsistency between components must be controlled, Easy to identify the presence of a watermark.
Multiplicative techniques: DCT, DFT, DWT	Transformations of cover audio that take place before a watermark is then added, allowing for different types of watermarking to take place	These are not watermarking techniques as such. Instead, they are a primary step to perform before the digitized audio is in a form that can then be watermarked.

Table 3.5: Advantages and disadvantages of watermarking schemes (contd..)

Technique	Advantages	Disadvantages
Spread spectrum	Watermark spread across spectrum so appears as noise, Robust technique, even against signal corruption, Can be embedded in significant components, which are more likely to survive attacks	Susceptible to corruption by compression, Requires the signal to remain synchronized.
Patchwork method	Can be implemented as a blind-decode system, Can be designed to be robust in given domains	Low capacity, Computationally complex.
Interpolation method	Good capacity, Resistant to Digital/Analogue conversion and transmission	Added 'virtual' components can cause perceptibility issues.

Advantages and disadvantages of the schemes presented in the tables - table 3.1, table 3.2 and table 3.3, are demonstrated through table 3.4 & table 3.5 (adapted from [Healy 2010]).

3.4 Summary

In this chapter, a thorough literature review has been carried out by grouping the techniques in different categories. Based on all these reviews combined with the problem set out in Chapter 1, it has been decided that an audio authentication scheme as well as a copy protection mechanism should be suggested, investigated and implemented.

The work will be carried out in the following sequence:

- Signal dependent feature vectors are identified and extracted and utilized for developing the FeatureMark for each audio signal;
- Prepared FeatureMark is embedded into the signal by estimating more precise frequency coefficients;
- An investigation on the possibility of embedding watermark bits based on manipulating the Walsh coefficients should be carried out;

Chapter 4

Data Collection and Feature Extraction

4.1 Introduction

This chapter focuses on the collection of speech data and how pre-processing is done to improve the result. Short-term processing of the signal manipulates sound inputs appropriately and helps in improving the results of analysis and synthesis. It also guarantees a better quality for the extracted watermark. Feature extraction is another important step that is described in this chapter where in some of the computational characteristics of the speech signal are mined for later investigation. Time domain signal features are extracted by employing FFT in Matlab. The features selected for this scheme are the physical features such as Mel-frequency cepstral coefficients, spectral roll-off, spectral flux, spectral centroid, zero-cross rate, short-time energy, energy entropy and fundamental frequency which directly correspond to the computational characteristics of the signal and are

not related to the perceptual characteristics.

4.2 Data Collection

Primary aim of this step is to collect speech data from different people. Around $6 \times 7 \times 10$ Malayalam speech signals are collected with duration of 50 seconds to 5 minute from 10 speakers. These signals include both male voices and female voices in '.wav' format. In all these recordings, speakers are asked to read the sentences in a normal voice. 6×7 stands for 6 samples of 7 different speech signals. Out of the 10 speakers we have selected, 5 of them are male and other 5 are female speakers. Thus a total of 420 speech signals were taken using music editor sound recorder.

Speech signals that are collected as part of this work include isolated words as well as sentences with varying signal duration. These are 'Ker-alam', 'Poojyam', 'Shadpadam', 'Vaazhappazham', 'Keralathilae mughya bhasha aanu Malayalam', 'Malayalam polae thannae pradhanappetta bodhana maadhyamam aanu English' and 'Tamizhu Kannada bhashakalkku ne-unapaksha bhasha padavi undu'. Recording signals at different time guarantees that speech trials are pronounced autonomously that of the preceding trials which make it more lenient to the variations in a person's voice that occur over short time duration.

4.3 Pre-Processing

Normally a speech signal is non-stationary but for short-time duration it appears to be stationary which results from the fact that glottal system cannot change immediately [Prasad and Prasanna 2008]. Therefore, in

speech processing it is often advantageous to divide a signal into frames to achieve its stationary nature.

According to the note presented in [Universitet 2004], a speech signal is typically stationary in windows of 20 ms. Therefore, this research the signal is divided into frames of 20 ms which corresponds to n samples (2.5):

$$n = t_s f_s \quad (4.1)$$

Framed signals have sharp edges towards its beginning and its end, hence it is essential to consider how to handle the sharp edges of these frames. Sharpness is induced by harmonics as edges add and can be handled using windowing techniques. Length of each frame is determined with time frame step and overlap. The time frame step, denoted by tf_s defines the duration between the start time of each frame. The overlap, denoted by t_0 define the interval from a new frame starts until the current stops. From this the frame length tf_l is depicted as:

$$tf_l = tf_s + t_0 \quad (4.2)$$

Hence each window is defined with a length tf_l which corresponds to $n_w = tf_l f_s$ samples and the reference to this study include [DeFatta, Lucas, and Hodgkiss 1995; Palani and Kalaiyarasi 2011; Ingle and Proakis 2011b; Leis 2002].

Framing and Windowing

Short-term processing is critical when working with voice signals and is performed to analyze individual frames.

Following figures - figure 4.1, figure 4.2 & figure 4.3 shows speech signal representations.

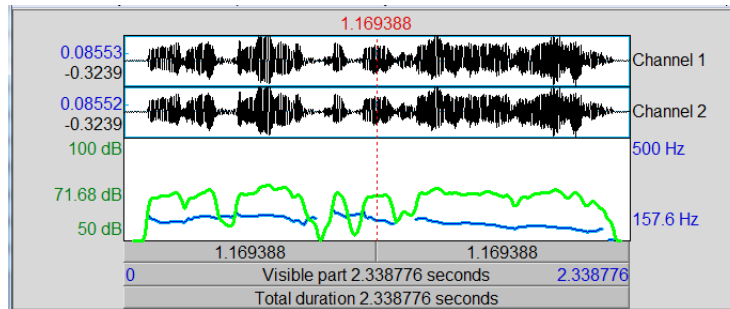


Figure 4.1: Speech signal - Waveform representation

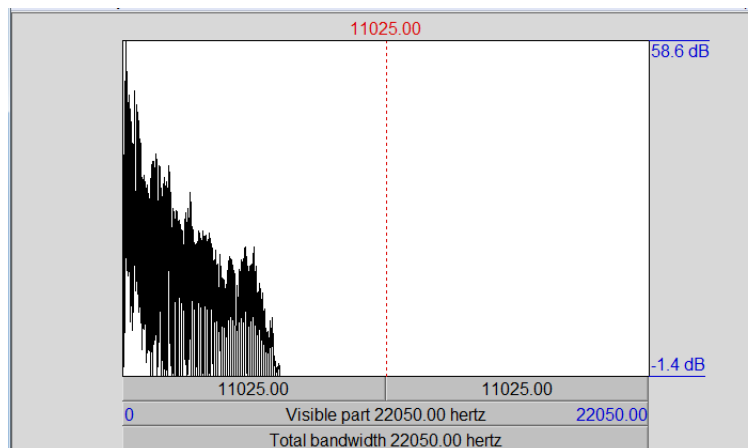


Figure 4.2: Speech signal - Spectrum

In any watermarking scheme, quality of the extracted watermark depends exclusively on the pre-processing action. Filtering process helps to remove noise content and alter frequency response characteristics of the signal in a preferred manner. Thus in this proposed scheme a frame based analysis is employed on speech signals [Universitet 2004; Quatieri 2002;

Wang 2004; Mathworks 1984; Arshad 2009; Chen 1993].

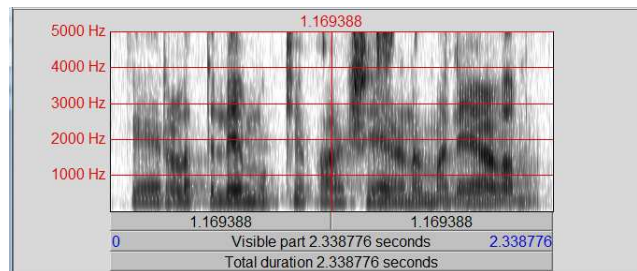


Figure 4.3: Speech signal - Spectrogram

Framing

In the proposed scheme, original speech signals are decomposed into a set of overlapping and non-overlapping frames.

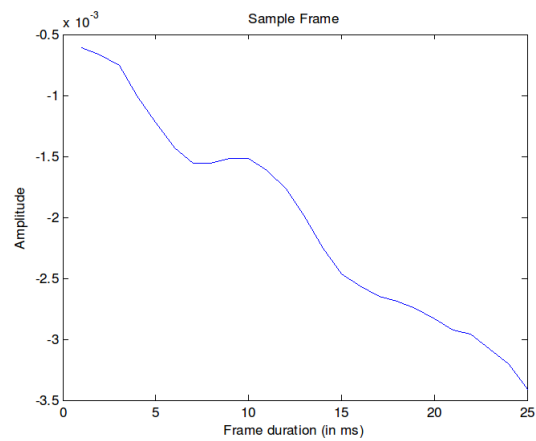


Figure 4.4: Single frame

This is done because the spectral evaluation of a signal is reliable if it

is stationary. That is, the region should be short enough for the signal characteristics to be uniform or approximately constant. To achieve this we have employed frames with duration of $10 \sim 25$ ms and with a frame rate of 100.

Windowing

Frames represented in figure 4.4 have sharp edges towards its start and its end. To tone down these edges Hamming windowing technique is employed using the Matlab's signal processing tool box. Here, speech segments are short-term estimates of the Fourier spectrum and each of these segments are effectively cross-multiplied with the Hamming window function to reduce the signal discontinuities at the edges of each frame.

Following figure 4.5 represents the results of applying a window function to frame.

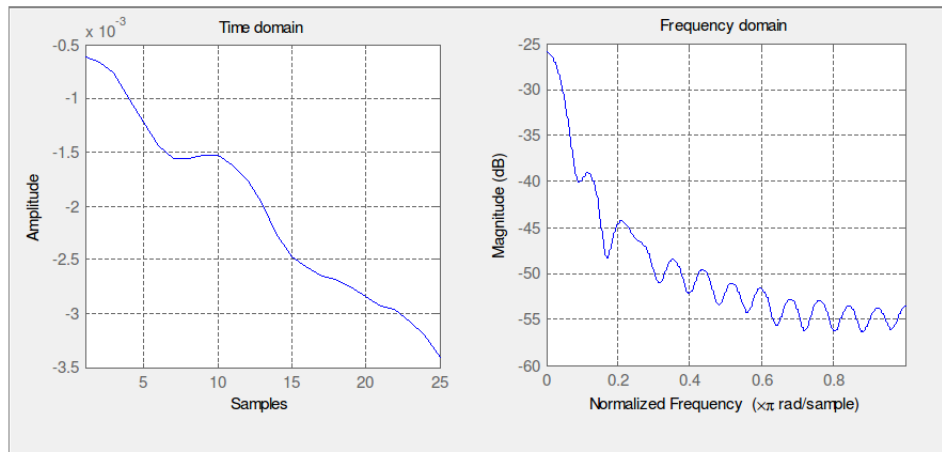


Figure 4.5: Single window

In the suggested scheme Hamming windowing technique is employed to reduce the high amplitude oscillations and thus provides a more gradual truncation to the infinite series expansion.

The generalized Hamming window can be represented as:

$$w[n] = \begin{cases} (1 - \alpha) - \alpha \cos \frac{2\pi n}{N-1} & , n = 0, 1, 2 \dots N - 1, \\ 0 & , \text{Otherwise.} \end{cases} \quad (4.3)$$

Hamming window scheme that employed in this system is a rectangular window function with an amplitude of 1 between $-Q$ and $+Q$ and the coefficients outside these windows are ignored. Wider this rectangular window, larger will be the value of Q .

Results of Hamming window function is represented in figure 4.6

$$w_H = \begin{cases} 0.54 + 0.46 \cos n \frac{\pi}{Q} & , |n| \leq Q, \\ 0 & , \text{Otherwise.} \end{cases} \quad (4.4)$$

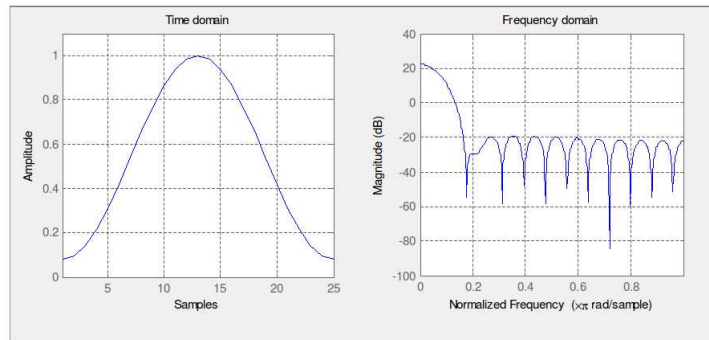


Figure 4.6: Hamming window

With windowing, frame part of the signal has been extracted without

added processing. Extracted frame whose frequency response has high side lobes help to barter the leakage energy from different and distant frequencies of these frames. This is done as the leakage as well as the periodicity occurred by the Fourier transformations are clearer in the frequency domain. A window function by itself inclines to have an averaging effect and thus it has low-pass spectral characteristics. Hamming windowing is performed in such a way that it preserves the spectral details as well as restricts the amount of spectral distortions aroused.

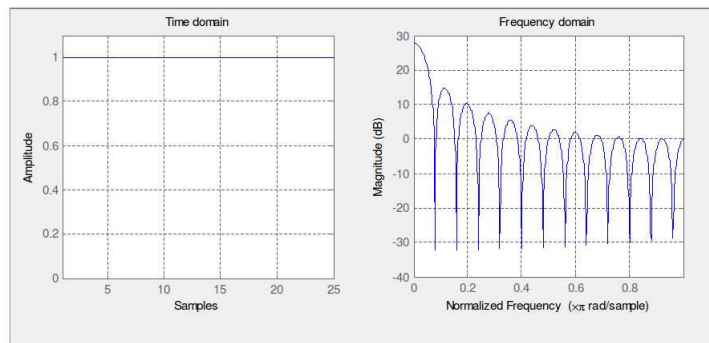


Figure 4.7: Rectangular window

Even though Hamming window has a wider main lobe, comparison between log magnitudes spectrum obtained for this Hamming window (represented in figure 4.6) and rectangular window (represented in figure 4.7) confirm the betterness of Hamming window that is employed. Following figure 4.8 illustrates a comparison between rectangular and Hamming windows.

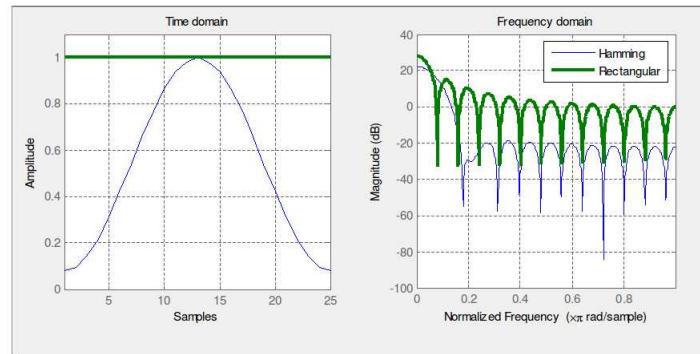


Figure 4.8: Comparison between rectangular and hamming windows

Thus windowing function helps to minimize the signal discontinuities towards beginning and end of individual frames. This is achieved by performing smooth transitions at these regions and hence attenuates the spectral leakage effects. Commonly used windowing techniques include Hamming windowing, Hanning windowing, Bartlett etc.. However, reducing spectral leakage in frequency spectrum of each frame by these windowing techniques may result in a modest loss of spectral resolution. Window functions usually reduce amplitude of the sidelobes but with a wider mainlobe resulting in a filter with lower selectivity. Thus, attention was given in selecting a window function that can reduce the sidelobes while approaching selectivity that can be achieved with rectangular window function. Width of the main lobe can be reduced by increasing width of the window i.e. the order of the filter.

Windowing function is designed by keeping the following aspects in mind:

- it should have a narrow band width main lobe

- it should have large attenuation in magnitudes of the sidelobes

A narrow main lobe helps to resolve the sharp details such as frequency response of the framed speech signal as the convolution continues in frequency domain and attenuated sidelobes helps to prevent noise from other parts of the spectrum from corrupting the true spectrum at a given frequency.

Frame shifting

Better temporal continuity in transform domain can be ensured with the use of overlapping windows and it is advisable to use an overlap of half or less the window size.

Frame Duration denoted by ‘N’ demonstrates the length of time over which a set of parameters are valid. For a speech signal the frame duration ranges between 10 ~ 25 ms. Frame Period denoted by ‘L’ specifies the period between successive parameter calculations and frame rate determines the number of frames computed per second. In this case the frame rate for short-term processing of speech signals is taken as 33 to 100. A Speech signal, its frames and feature vectors can be understood from the figure 4.9.

Upon completion of pre-processing tasks; de-framing is applied. De-framing is the process that combines these individual frames into a comprehensible speech signal. The two aspects that are considered includes:

- to take account of the previously used window
- to combine the samples shared by different frames

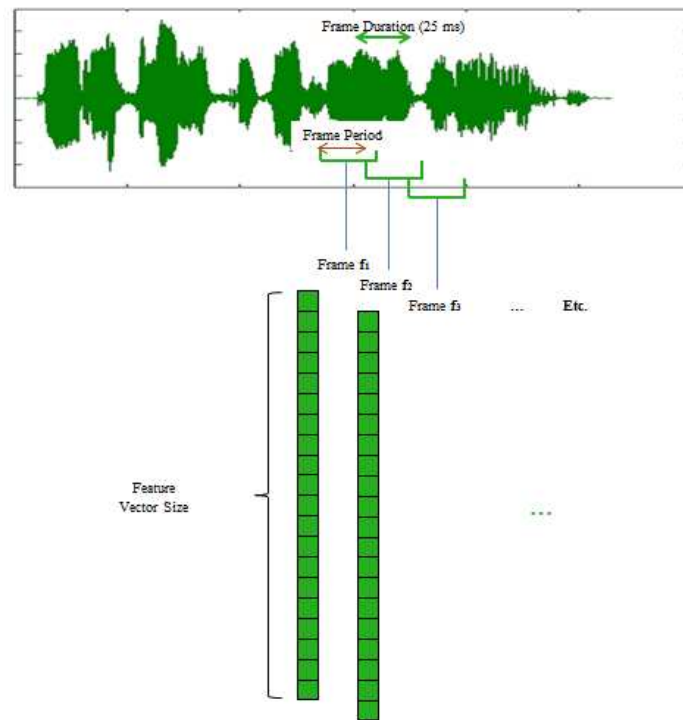


Figure 4.9: Representation of a speech signal, its frames and the feature vectors

For this process, each frame is multiplied by the reciprocal window by considering the fact that the window is different from zero. If two different frames have common samples, the frames are combined by taking the average of these samples such that closer the samples get to the edge of the frame less weight they are given [Chang, Wang, and Shen 2010; Lei and Soon 2012].

4.4 Feature Extraction

Analysis and characterization of an audio content is performed by audio feature extraction [Mierswa and Morik 2005; MAT 2009; Lartillot and Toivainen 2007]. Some applications that need well-organized feature extraction are auditory scene analysis, steganography and watermarking, content-based retrieval, indexing and fingerprinting. As mentioned in [Umaphathy, Krishnan, and Rao 2007], the key to extract strong features that characterize the complex nature of audio signals is to identify their discriminatory subspaces [Kameoka, Nishimoto, and Sagayama 2005; Liu, Wang, and Chen 1998; Tzanetakis 2004; Mathieu et al. 2010; McKay, Fujinaga, and Depalle 2005; Bullock and Conservatoire 2007; Lartillot and Toivainen 2007].

Speech signals which are one-dimensional in nature are represented as time series in a two-dimensional plane. The plot takes amplitude or intensity (in decibel dB) towards its y-axis and conforming time interval (in seconds or milli seconds) towards the x-axis. Feature extraction module deals with extracting features that helps to differentiate each member in the communicating group and store these feature values in the database.

Typical procedure of a feature extraction module can be depicted as follows: figure 4.10:

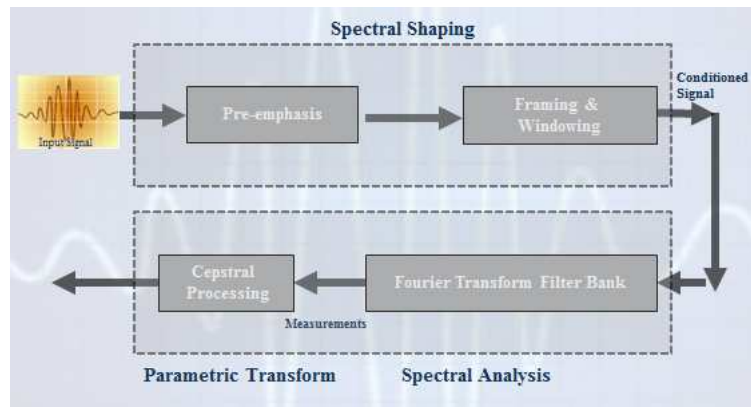


Figure 4.10: Feature extraction

Figure 4.10 indicate that, first phase in feature extraction is pre-emphasis. Pre-emphasis filter is a high pass filter that helps to boost the signal spectrum approximately to 20 dB. It helps in preventing the numerical instability of voice spectrum and keeps low frequency components from dominating the spectral envelope and hence obtain similar amplitude for all formants.

In the proposed system, feature extraction is done by performing Fourier transform on the signal, where Fourier analysis decomposes the sampled signal into its fundamental periodic components such as sines and cosines. An existing Fast-Fourier transformation such as Cooley-Turkey algorithm maps the given time space into its corresponding frequency space. Transforming the signal from its time domain to its frequency domain is important in context of audio signals [Mathworks 1984].

As discussed in chapter 2, physical features of an audio signal are extracted and analyzed so as to employ it in applications like classification and information hiding (steganography and watermarking) . Intention of this module is to extract the features that are invariant to irrelevant trans-

formations and have good discriminative power across different classes of signals. Numerical representations of each acoustic signals termed as the feature values are extracted and used in classification as well as in the generation of watermark employed in the proposed schemes.

Computable characteristics of the time domain signals which are not related to human perception are extracted and used. Features employed include Mel-frequency cepstral coefficients, spectral flux, spectral-roll off, spectral centroid, energy-entropy and short-time energy which characterize the low-level or reduced dimension parameters and thus stand for specific temporal and spectral properties of the signal.

4.4.1 Mel-Frequency Cepstral Coefficients (MFCC)

The Mel-frequency cepstral coefficients are restricted to 20 which results in a data matrix of 20×256 coefficients. Embedding such a huge data volume in audio will expose furtiveness of the system; hence vector quantization is implemented resulting in a data matrix of 20×1 coefficients. Steps involved in extracting the MFCC values are depicted using the following figure 4.11 :

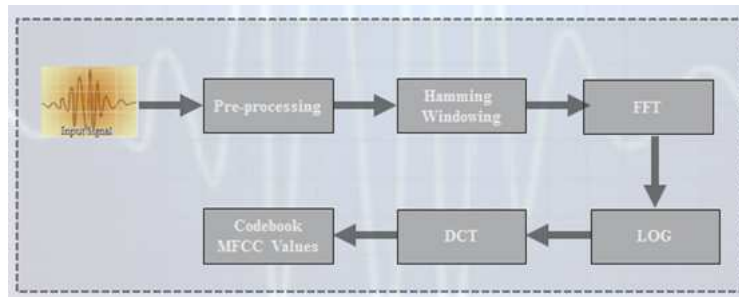


Figure 4.11: MFCC feature extraction

Towards the extraction of MFCC values, a codebook function for each input signal is performed. This function opens corresponding speech signal mentioned in 'sound.txt' and generate a codebook of size 13×16 . Size of the codebook can be adjusted by altering the dimension mentioned in the procedure. For each FFT bin, exact position in the filter bank is identified to find the original frequency response which is preceded by the inversion of filter bank center frequencies. Then identify the integer and fractional sampling positions. Subsequently, actual processing is started, in which each chunk of data is windowed with a Hamming windowing, shift these windows into an FFT order and calculate magnitude of the FFT. FFT data obtained are converted to corresponding filter bank outputs and then find its base 10 log values which is processed with the cosine transformation in order to reduce its dimensionality. Accuracy of this evaluation procedure is done by reconstruction of the original filter bank outputs. It involves multiplying cepstral data by transpose of the original DCT matrix. Once it is done, FFT bins are combined to make the original signal and it works as expected because the DCT matrix is carefully scaled to be orthonormal.

Mel-frequency cepstral coefficients extracted for a sample time domain signal can be depicted as follows - figure 4.12 :

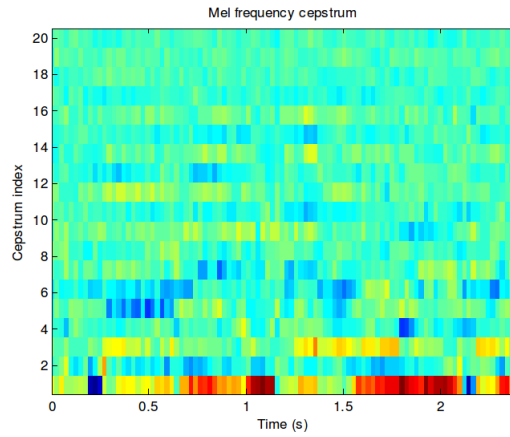


Figure 4.12: Mel-cepstrum in time domain

A graphical view of the above mentioned vector quantized 20×1 coefficients can be presented as follows - figure 4.13 :

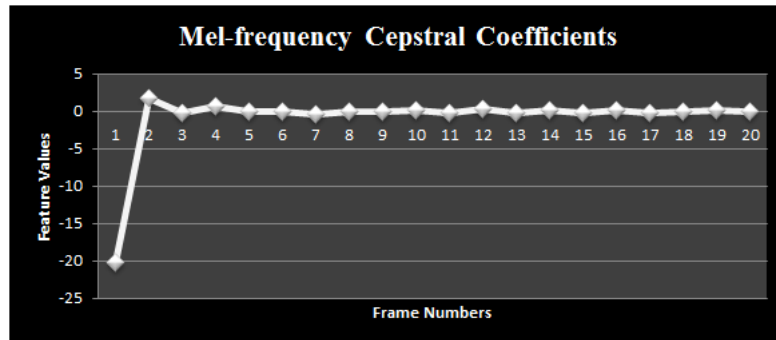


Figure 4.13: MFCC graph

The values obtained for each signal is in an order of 20×1 and a sample set of values are listed in the following table 4.1

Table 4.1: MFCC values

-20.16315	1.73489	-0.27973	0.70411	-0.11126
-0.01661	-0.34455	-0.01347	0.01147	0.21281
-0.17715	0.27118	-0.15787	0.15647	-0.20811
0.18670	-0.16586	-0.00982	0.09074	-0.03288

4.4.2 Spectral Flux

Along with the evaluation of MFCC values, the spectral flux is also evaluated for each window which results in second normal form of the obtained spectral amplitude difference vector.

Let $f(i), f(i + 1), f(i + 2) \dots f(i + n)$ be the frames resulted due to pre-processing activity. Then evaluate the spectral amplitude difference of first two frames represented as $|f(i) - f(i + 1)|$ and then take its second normal form in order to calculate its spectral flux values. Its evaluation also employs FFT.

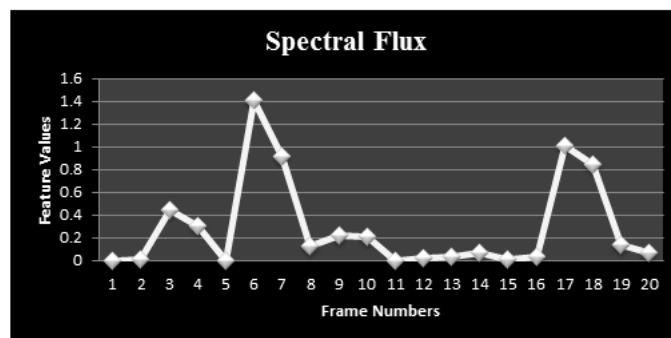


Figure 4.14: Spectral flux feature graph

In the proposed scheme, the number of frames depends on the signal

duration. For signals with less duration, number of frames will be less and as the signal duration increases, number of frames also increases. Spectral flux values are generated for each frame and the values are more or less related to each other. The values obtained (represented in figure 4.14) for a male speech signal is shown below (Table 4.2) :

Table 4.2: Spectral flux values

0.00000	0.00580	0.44065	0.30889	0.00312
1.40919	0.91428	0.12775	0.21963	0.20456
0.00118	0.02591	0.03857	0.06399	0.00565
0.02903	1.00418	0.84683	0.14013	0.07017

A 20×1 matrix is obtained with 20 numbers of frames.

4.4.3 Spectral Roll-Off

Spectral roll-off, as defined in chapter 2 is estimated for each window resulting in a matrix of order $1 \times N$; where N corresponds to the total number of frames/windows. As the spectral-flux evaluation, calculation of roll-off also utilizes FFT. Exact position of filter bank is identified for each FFT bin which helps to obtain the original frequency response of that bin. This step is preceded by inversion of filter bank center frequencies. After retrieving the original frequency response, its integer and fractional sampling positions should be identified. Then each chunk of data is windowed with a Hamming windowing, shift these windows into an FFT order and calculate the magnitude of the FFT. Applying the equation 2.7 on each of the windows retrieve the spectral roll-off values for that window.

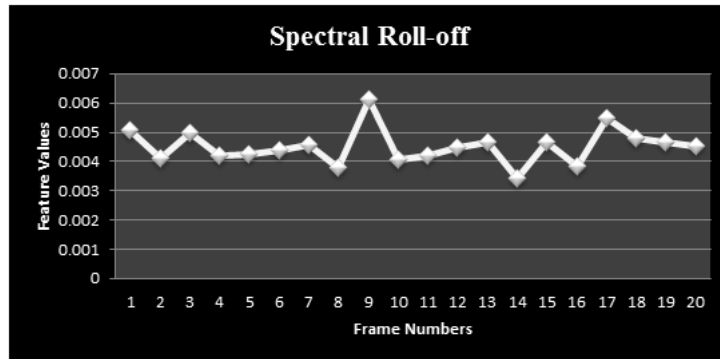


Figure 4.15: Spectral roll-off feature graph

Spectral roll-off values (represented in figure 4.15) obtained for the same acoustic signal is given below:

Table 4.3: Spectral roll-off values

0.00508	0.00413	0.00499	0.00422	0.00426
0.00440	0.00458	0.00381	0.00612	0.00408
0.00422	0.00449	0.00467	0.00340	0.00467
0.00385	0.00549	0.00481	0.00467	0.00454

A 1×20 matrix is obtained as the number of frames is 20.

4.4.4 Spectral Centroid

Evaluation of spectral centroid for each signal reveals the balancing point of spectral power distribution. Centroid values are also identified with the help of FFT calculation. In this case also, the original frequency response is evaluated for each FFT bin by identifying the exact position in filter bank. Each chunk of data is then windowed with a Hamming windowing.

Magnitude of FFT is calculated by shifting these windows into an FFT order. Applying the equation 2.6 on each of the window retrieves the centroid values for that window.

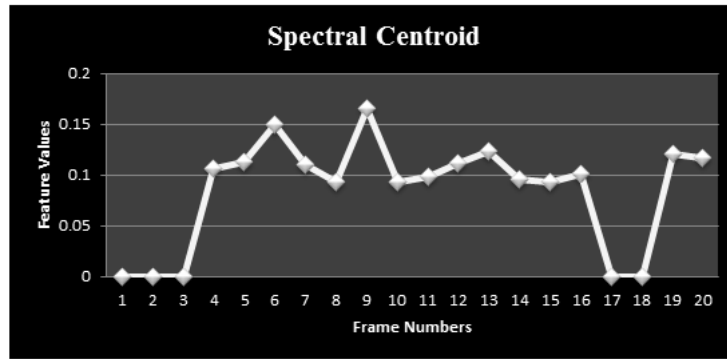


Figure 4.16: Centroid value plot

Centroid values (represented in figure 4.16) obtained for each of the frames in the matrix of order 20×1 are shown below:

Table 4.4: Centroid values

0.00000	0.00000	0.00000	0.10693	0.11292
0.15038	0.11033	0.09305	0.16497	0.09273
0.09856	0.11135	0.12376	0.09553	0.09317
0.10090	0.00000	0.00000	0.12043	0.11649

4.4.5 Energy entropy

Energy entropy feature is calculated for each frame by splitting it into sub-frames having fixed duration. Exact position in the filter bank is identified for each FFT bin which helps to find the original frequency response and

then to identify the integer and fractional sampling positions. Shift the obtained Hamming windows into an FFT order and then calculate its magnitude. Applying the equation 2.8 on each of the window retrieves entropy values for that window. From these values, a normalized value is obtained for each window.

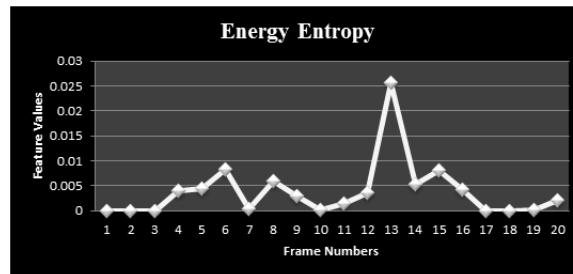


Figure 4.17: Entropy feature plot

Energy entropy feature values (represented in figure 4.17) obtained for a signal which was segmented into 20 frames are shown in the following table 4.5 with an order of 1×20

Table 4.5: Entropy values

0.00000	0.00000	0.00004	0.00405	0.00437
0.00825	0.00049	0.00599	0.00292	0.00013
0.00148	0.00353	0.02572	0.00526	0.00814
0.00429	0.00001	0.00005	0.00021	0.00220

4.4.6 Short-Time Energy

Short-Time Energy which distinguishes voiced speech from unvoiced speech evaluates the amplitude variation for each of the Hamming window. Num-

ber of values obtained is directly proportional to the number of frames or windows generated for handling the signal. In this case also, the original frequency response is evaluated by identifying the exact position in the filter bank for each FFT bin. This step is preceded by the inversion of filter bank center frequencies. Identification of its integer and fractional sampling positions is followed by actual processing in which each chunk of data is windowed with a Hamming windowing, shift these windows into an FFT order and calculate the magnitude of the FFT. Applying the equation 2.9 on each of the window retrieves the short-time energy values for that window.

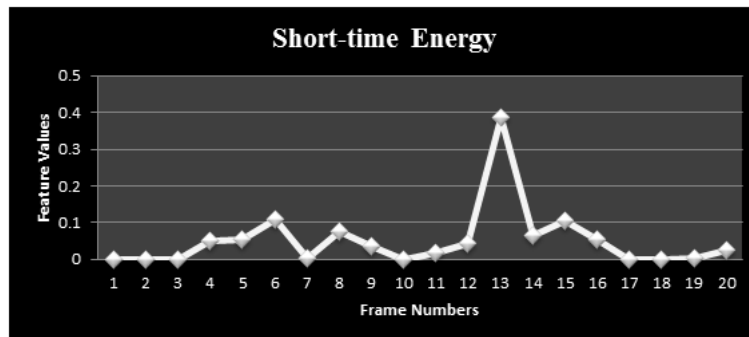


Figure 4.18: Energy plot for a single frame

Short-time energy values (represented in figure 4.18) obtained for a signal which is segmented into 20 frames are shown in the following table 4.6 with the order 20×1 .

Table 4.6: Energy values

0.00001	0.00001	0.00037	0.04873	0.05261
0.10870	0.00482	0.07459	0.03635	0.00112
0.01594	0.04269	0.38620	0.06506	0.10421
0.05347	0.00004	0.00039	0.00192	0.02602

4.4.7 Zero-Cross Rate

Measure of dominant frequency or the number of time domain zero-crossings within a speech frame is calculated as the rate of sign changes along each frame of the signal.

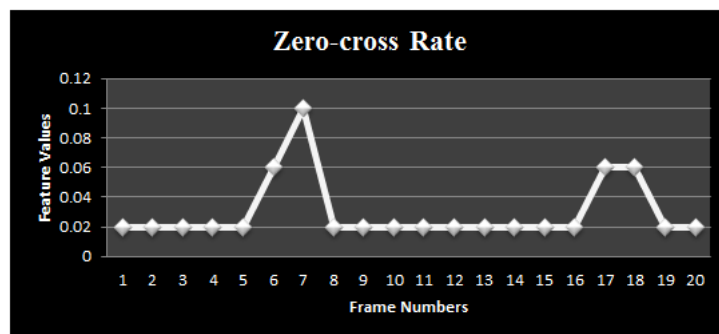


Figure 4.19: Plot of ZCR values

In order to evaluate the ZCR values, the first step identifies the original frequency response of the signal. For this, first we need to invert the filter bank center frequencies then identify the exact position in the filter bank for each FFT bin. Thereafter, identify integer and fractional sampling positions which is followed by windowing of each chunk of data. Obtained windows are shifted into an FFT order and calculate the magnitude of FFT.

Applying the equation 2.5 on each of the window retrieves the zero-cross rate values for that window.

Speech signal is divided into 20 frames and zero-crossings obtained is in the matrix of order 20×1 . Sample results (represented in figure 4.19) obtained are given below:

Table 4.7: Zero-crossing values

0.02000	0.02000	0.02000	0.02000	0.02000
0.06000	0.10000	0.02000	0.02000	0.02000
0.02000	0.02000	0.02000	0.02000	0.02000
0.02000	0.06000	0.06000	0.02000	0.02000

4.4.8 Fundamental Frequency

Fundamental frequency is measured by taking the inverse of its period (T) and its value is evaluated as 44,100 Hz in the suggested scheme. That is, it is the inverse of the minimum interval on which the signal repeats.

$$f_0 = \frac{1}{T} \quad (4.5)$$

4.5 Summary

From this chapter, the pre-processing tasks such as framing and windowing that are performed on each of the recorded speech signals are identified and a detailed explanation on what have been done towards it. The number of frames considered is 20 and correspondingly the number of samples. It entirely depends on the signal duration. This chapter also describes the feature extraction module which demonstrates the features employed or extracted for the use of entire scheme and how each of these features are

retrieved from the signal. The features that are utilized in this scheme are the signal dependent physical features and are used in the development of the watermark.

Chapter 5

Speaker Recognition - Verification and Identification

5.1 Introduction

This chapter deals with the identification of features that help in recognizing the speakers participated in the communication scheme. Feature vectors extracted in the feature extraction module are verified using three different classifiers such as ANN, k-NN and SVM . Best features identified in this module are utilized towards the generation of watermark for the proposed schemes. Speaker recognition itself is a challenging area of research and an effort has been taken to identify some signal dependent features that helps in this recognition task.

5.2 Speaker Recognition

The most accepted form of identification for human is his speech signal. The speaker recognition process based on a speech signal is treated as one of the most exciting technologies of human recognition [Orsag 2010]. As prescribed in Chapter 2, audio signal features can be classified either in the perceptual mode or in the physical mode. In the proposed work, we have mainly employed the physical features towards the speaker identification activities. Speaker identification and speaker verification are the two schemes that are part of a speaker recognition task. As discussed in [Orsag 2010], the process of the speaker identification answers a question “Who is speaking?”. On the other hand, the speaker verification answers a question “Is the one, who is speaking, really the one, who he is claiming to be?”. The speaker recognition can be done in several ways and the commonly used method is based on the hidden markov models with the gaussian mixtures (HMM-GM) [Baggenstoss 2008]. But the proposed scheme employs ANN, k-NN and SVM classifiers.

Following figures 5.1 & 5.2 illustrate signal representations of male and female voices for the utterance of same sentence.

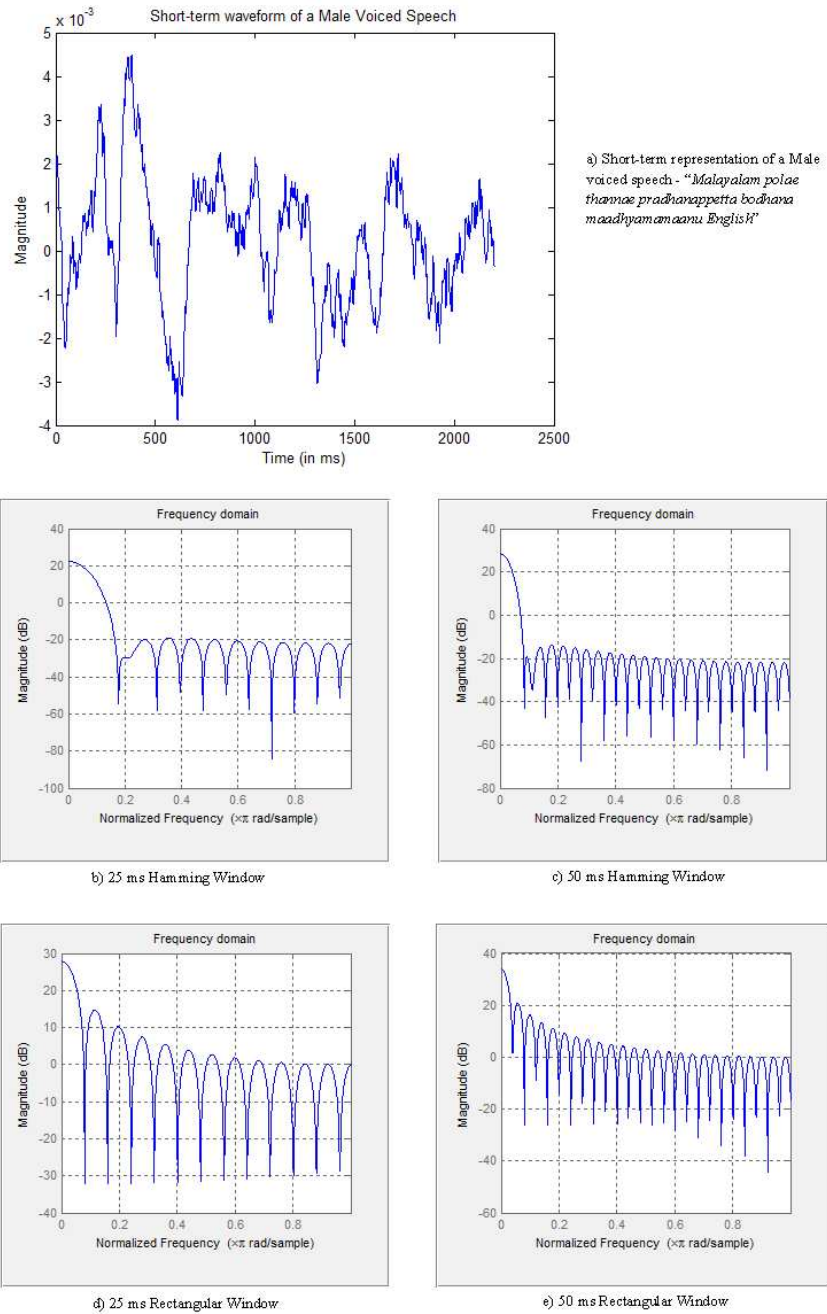


Figure 5.1: Male voiced speech

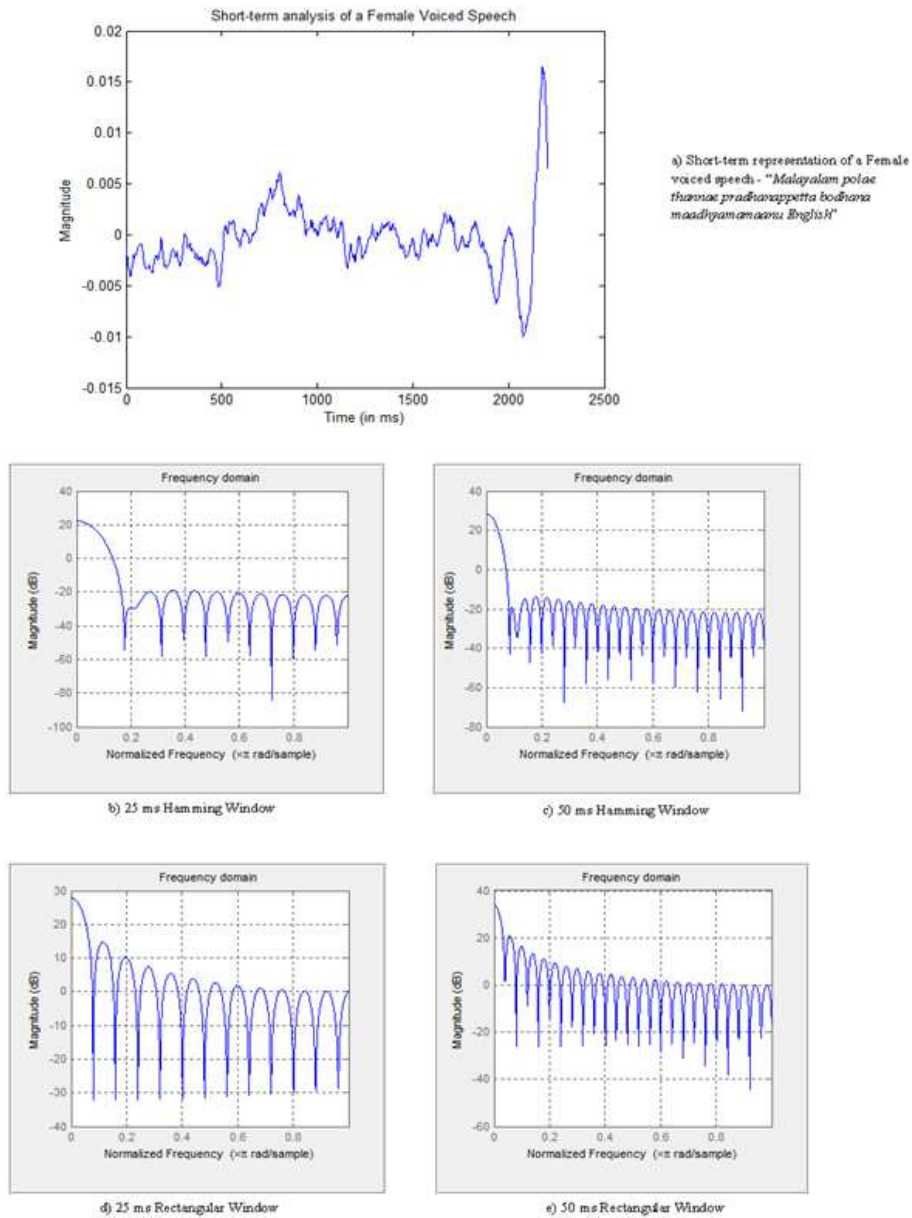


Figure 5.2: Female voiced speech

5.3 A Brief Review of Literature

As discussed earlier, speaker recognition itself is a broad research area which comprises of speaker verification and speaker identification. Aim of this section is to brief out some of the works conducted in the field of speaker recognition.

In 2010 Filip Orsag [Orsag 2010] created a new speaker dependent feature called speaker dependent frequency cepstrum coefficients (SDFCC) for the purpose of speaker recognition. This scheme employs the speaker recognition technology based on the HMM with the newly generated feature sets (SDFCC). Even if these coefficients aim at the speaker recognition activities, in some special cases these may be usable for the speech recognition. SDFCCs differ from the MFCCs is in the utilization of a separate filter bank. In the paper [Pathangay and Yegnanarayana 2002], a text dependent audio-visual biometric person authentication scheme is suggested using dynamic time warping (DTW). This scheme uses a combination of features extracted from video and audio such as the mid-face vertical intensity and the LPC coefficients respectively.

The work presented in [Kavitha 2013] introduces an automatic speaker recognition system which works with the Matlab toolboxes and the feature employed towards this scheme is the MFCC. Douglas in his paper [Reynolds 2002] provides a brief overview of the area of speaker recognition, which includes its applications, various techniques used to achieve this, performance evaluations, its strengths and weaknesses and finally some future scope in this field. In [Peacocke and Graf 1990], give a description on speech as well as speaker recognition schemes. They stated that, speaker recognition is related to speech recognition. When the task involves identifying the person talking rather than what is said, the speech signal must be processed

to extract measures of speaker variability instead of being analyzed by segments corresponding to phonemes or pieces of text one after the other. In his paper [Melin 1999], suggests an overview of the activities in this working group (WG) . WG2 of the COST250 Action “Speaker Recognition in Telephony” has dealt with databases for speaker recognition. First results demonstrate an overview of 36 existing databases that has been used in speaker recognition research. Second result is the publicly available Poly-cost database, a telephony-speech multi-session database with 134 speakers from all around Europe.

[Ghosh et al. 2004] suggests a speaker recognition scheme which employs artificial neural network for the purpose of identifying the speakers participated in the communication by employing features such as MFCC and LPC. The work presented in the thesis [Feng 2004], introduces a new method to combine the pitch information with MFCC features for identifying the similarity in k-NN algorithm. This scheme improves the speaker pruning accuracy. Experiments were conducted on HMM modeling and recognition with different setups. A comparative study is conducted on the error rate obtained with and without speaker pruning. The work suggested in [Stolcke and Ferrer 2012], demonstrate that a speech recognizer trained on full-bandwidth, distant-microphone meeting speech data yields reduced speaker verification error for speaker models based on MLLR features and word-N-gram features. [Ferrer et al. 2011] introduces a new database created using data from NIST SREs from 2005 to 2010 for evaluation of speaker recognition systems. As the paper suggest, this database involves types of variability already seen in NIST speaker recognition evaluations (SREs) like language, channel, speech style and vocal effort and new types not yet available on any standard database like severe noise and reverberation.

The work presented by [Ellis 2001], entails the design of a speaker recognition code using Matlab. Speech signals are handled by analyzing its time and frequency domain and using a 3rd order Butterworth filter removes the background noise to a great extent. In his thesis [Jin 2007] focuses on improving the robustness of speaker recognition systems on far-field distant microphones. The work is performed in two dimensions. First, they have investigated approaches to improve robustness for traditional speaker recognition system which is based on low-level spectral information and in turn introduces a new reverberation compensation approach. Second, they have investigated approaches to use high-level speaker information to improve robustness which in turn introduces techniques to model speaker pronunciation idiosyncrasy from two dimensions: the cross-stream dimension and the time dimension. The documentation given in the paper [Biometrics.gov 2006], provides a better understanding in the field of speaker recognition - it discusses a brief introduction, some history behind it and the approach employed towards the speaker recognition tasks. In the paper [Hennebert et al. 2000], presents an overview of the POLYCOST database dedicated to speaker recognition applications over the telephone network. This paper describes the main characteristics of this database such as medium mixed speech corpus size (>100 speakers), English spoken by foreigners, mainly digits with some free speech, collected through international telephone lines, and minimum of nine sessions for 85% of the speakers. In their work presented in [Barlow, Booth, and Parr 1992], created two speech databases intended for the purpose of speaker recognition which includes a very large laboratory access database and small departmental database. The aim of this work was to overcome the drawbacks of existing speech databases towards the speaker recognition activities. In the work presented by [Kinunen 2005], a better text-independent speaker recognition strategy has

been discussed which employs better features or better matching strategy or a combination of the two towards its recognition tasks.

[Liu, He, and Palm 1996] in their work reveals that among the various parameters such as pitch, LPCC, $\delta LPCC$, MFCC, $\delta MFCC$ that are extracted from speech signals, LPCC and MFCC as well as a combination of these features with pitch are effective representations of a speaker and thus help in the recognition tasks to a great extent. In the paper [Carey et al. 1996] demonstrates the importance of prosodic feature in the recognition process and its performance is measured using HMM models. Gaussianization, another important process performed for the speaker verification is presented in [Xiang et al. 2002] by employing the gaussian mixture models (GMM). A viable alternative to GMM systems is proposed by [Zilca 2001], functions through two verification methods, namely, frame level scoring and utterance level scoring. Studies of [Yu, Mason, and Oglesby 1995] compares HMM, DTW and VQ for speaker recognition and identified that for text-independent speaker recognition, VQ performs better than HMMs and for text-dependent speaker recognition also DTW outperforms VQ and HMM based methods. Another technique proposed by [Inman et al. 1998] derives the segment boundary information from HMMs [Russell and Jackson 2005] which in turn provides a means of normalizing the formant patterns. An HMM based text-prompted speaker verification system was suggested in the work presented in paper [Che, Lin, and Yuk 1996]. Employing an adaptive vocal tract model that emulates the vocal tract of the speaker towards the speaker recognition task was presented by [Savic and Gupta 1990]. A New set of features termed as adaptive component weighting (ACW) cepstral coefficients introduced by Khaled T Assaleh are utilized for speaker recognition presented in [Assaleh and Mammone 1994]. The papers [Linghua, Zhen, and Baoyu 2004; Pelecanos et al. 2000] introduces

the work that are intended for improving the performance of the existing schemes that works through VQ based gaussian modelling and training VQ codebook for HMM-based speaker identification. In the work [Rao et al. 2007], introduces a text-dependent speaker recognition system for Indian languages by creating a password authentication system.

In the paper [Kinnunen and Li 2010], provides a brief overview of the speaker recognition technology which comprises of the classical and state-of-the-art methods, recent developments and the evaluation methodologies adopted. [Garcia-Romero and Espy-Wilson 2011] introduces a method that helps to boost the performance of probabilistic generative models that work with i-vector representations in speaker recognition process. In the article [Sahidullah and Saha 2012], presents a novel feature extraction scheme that captures complementary information to wide band information. These features are tested on NIST SRE databases that work with GMM and obtained good performance results.

Many works have been reported in the field of speaker recognition. Out of these, few papers are collected and details of the works presented are reported over here.

5.4 Verification Process

General outline of the existing speaker verification schemes can be represented as in the following figure 5.3 [Reynolds 2002]:

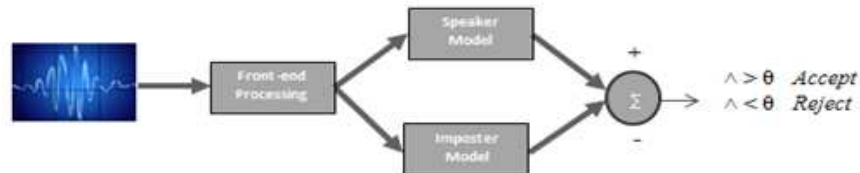


Figure 5.3: Speaker verification process

The hypothesis that needs to be tested is: whether the test speech comes from the claimed speaker or from an imposter. In this process, features extracted from the speech signal are compared to a model representing the claimed speaker to a previous enrollment and also to some models of the imposter. Likelihood ratio statistic decides whether to accept or reject the speaker where the ratio represents the difference in log domain of speakers.

General techniques as outlined in figure 5.3 use three main components such as front-end processing, speaker models and imposter models in speaker recognition tasks. Front-end processing also termed as feature extraction is described in detail in the former chapter 4. Next is the speaker modeling that incorporates a theoretical foundation that helps in understanding the model behavior and the mathematical evaluations. Then generalize already enrolled data which helps to fit the new data appropriately and finally gives a mean representation of data in both size and computation.

Speaker modeling techniques that are employed in the proposed scheme include the nearest neighbor, neural networks and support vector machine models. In ANN technique all feature vectors from the designated speech are retained to characterize the speaker. The ANN methods used have

different methods like multi-layer perception or radial basis functions. The key difference in this model is that it utilizes explicit training to differentiate the speaker being modeled and other speakers. The drawback to this model is that training is computationally costly and cannot be generalized. In the case of k-NN classifier, with verification a match score is derived as the cumulated distance of respective feature vector to its k nearest neighbors in the speaker's training vectors. Support vector machine doesn't need an explicit model instead all feature vectors from the designated speech are retained to represent the speaker.

Third step is the imposter modeling which is critical for good performance results. It basically acts as a normalization to help minimize non-speaker related variability (e.g., text, microphone, noise) in the likelihood ratio score as presented in [Reynolds 2002]. First approach in this process is to calculate the imposter match score which is a function of maximum or average of the scores for a set of imposter speaker models. Second approach uses a general speaker-independent model that is to be compared with a speaker-dependent model. General model provides better performance than the first cohort scheme, as it allows the use of maximum A-posteriori (MAP) training to adapt the claimant model from the background model, which can increase performance and decrease computation and model storage requirements.

5.5 Speaker Recognition with Artificial Neural network

Artificial neural networks (ANN) are computational models inspired by animal central nervous systems that are capable of machine learning and

pattern recognition. They are usually presented as systems of interconnected “neurons” that can compute values from inputs by feeding information through the network [Encyclopedia 2013; Science and Engineering Laboratories 2010]. Many works have been reported in the field of speaker recognition using ANN models [Von Kriegstein et al. 2005; Syrdal and Gopal 1986; Von Kriegstein and Giraud 2006; Neto et al. 1995; Hambley 2008].

Neural network classification phase in this scheme needs a large set of feature data that are extracted in the feature extraction module from a diverse collection of recorded speech signals. ANN speaker recognition module has been conducted with the use of signal processing toolbox in Matlab.

5.5.1 Training Data

In this work, a total of $6 \times 7 \times 10$ Malayalam speech signals are selected for the use as training data for ANN network. 10 speakers, 5 males and 5 females volunteered to individually record the signals at different times. These signals include words as well as sentences with varying signal duration. These include ‘keralam’, ‘poojyam’, ‘shadpadam’, ‘vaazhappazham’, ‘keralathilae mughya bhasha aanu malayalam’, ‘malayalam polae thannae pradhanappetta bodhana maadhyamam aanu english’ and ‘tamizhu kannada bhashakalkku neunapaksha bhasha padavi undu’. Classification is done with the data from five male and five female speakers with 6 samples of each on the 7 words and/or sentences uttered. The ANN network trained is more robust and be able to differentiate between speakers of both sexes with related precision. Recording signals at different time guarantees that speech trials are pronounced autonomously of the preceding trials which

make it more lenient to the variations in a person's voice that occur over short time duration.

Thus we have collected a total of 420 initial sound samples using the music editor sound recorder. Randomly selected utterances of each word from every person (109 training samples) were used as training data. The other two samples (59 testing samples) were used for validation and test purposes in the networks. Speaker modeling techniques performed confirms the signal characteristics that aid in speaker recognition. In perspective of the proposed watermarking schemes the samples collected holds good though more number of samples improves the accuracy of classification.

On obtaining the pre-processed sound samples for training purpose, `nprtool` in Matlab had been executed. Command `nprtool` open the neural network pattern recognition GUI. It is characterized by a hidden layer, logistic activation functions and back propagation algorithms. In order to test different combinations of data and features, the data sets are also grouped in different combinations and saved as different files. A total of 28 different data sets have been created with different combinations of speakers and features extracted from each of these samples. Values in the matrix or file is then arranged such that each speaker had the same number of training samples which helps to avoid the biasing of network towards any speaker. Rests of the values were utilized towards the validation and testing purpose. Also, the training data were arranged randomly to ensure that the network would not be trained on a long sequence that corresponds to any one speaker's signal consecutively which in turn guarantees the training of the network more evenly among speakers.

Initially, four different types of inputs were used, as shown in Table 5.1, with six speaker combinations for each, as presented in Table 5.2.

Table 5.1: Types of inputs (420 inputs signals of 10 members)

20 MFC coefficients (for a signal)
20 centroid values (for a signal)
20 short-time energy values (for a signal)
20 energy-entropy values (for a signal)
20 spectral flux (for a signal)
20 spectral roll-off (for a signal)
20 zero-cross rate (for a signal)

Table 5.2: Types of speech files (5 male and 5 female speakers)

Isolated malayalam words
Continuous malayalam speeches

Likewise, total of $6 \times 7 \times 10$ data files have been employed, each used to train a separate network. This diversity should give an accurate assessment of the proposed algorithm's ability to perform under different situations.

5.5.2 Testing Data

After completion of proper network training, next phase is testing the network with two types of data that falls under text-dependent and text-independent speaker recognition schemes. This is to compare the ability of the network to differentiate the speakers who have participated in the communication for these two types of data items.

As the first step, network is tested with randomly selected data sets for each of the words or sentences used in training. In an ideal case, the network should be able to recognize the speaker with maximum probability if he/she speaks a word or sentence used in training.

5.5.3 Experimental Results

Experiments were conducted based on a voice dataset and performance of the retrieved features were tested based on this. Voice dataset used for all the experiments that were conducted as part of this scheme was gathered from fellow research team members and colleagues through face to face interaction. This exercise gave a wide set of sample speech signals that are more or less similar to each other.

Voice database consists of 10 speakers, 5 males and 5 females. All speakers were in the age group 28 - 38 and collected a total of 6 samples for each 7 words/sentences they uttered. The speaker age group by virtue got restricted to 28 - 38 as the voice dataset gathered was from fellow research team members and colleagues through face to face interaction. Speakers age or the spoken language does not impact accuracy of the result set due to the fact that proposed schemes evaluate the physical signal characteristics of the respective speakers which is independent of the speaker age or spoken language. Recorded samples were used to test the quality of derived features towards speaker recognition. Speech samples in '.wav' file format were recorded using the music editor sound recorder with a low signal-to-noise ratio. Sampling frequency for the recordings was 44,100 Hz with a precision of 16-bits per sample.

Feature Sets

Testing is conducted independently with individual feature sets for all selected samples and with a selected combination of 2 or more feature sets. First feature sets consists of 20 MFCC values or other feature values and the second type of feature sets includes different combinations of features

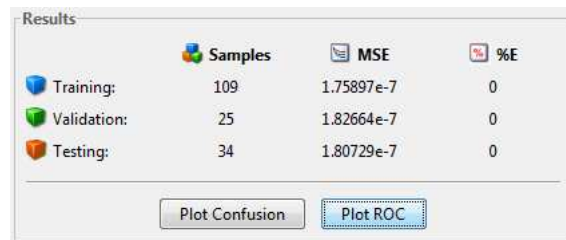
that are presented in table 5.1 for each speech samples. Feature sets selected for the training as well as the testing schemes varies depending upon the type or types of features selected in the process of classification.

Evaluation Using Feature Sets

In an experiment that have been conducted, evaluation of speaker recognition based on MFCCs are performed by employing 109 samples in training data set each with 20 MFCC values, 25 samples in validation data set and the rest 34 samples in test data set. In the four members, 2 of them are male speakers and the other two are female speakers. On multiple training, different results have been obtained; the reason behind this is that, its initial conditions and sampling for each iteration.

From this set, a particular iteration takes 27 samples of the first member, 29 samples of the second member, 26 samples of the third member and 27 samples of the fourth member in the training data set. That is, these are the data used in training algorithm to compute the adjustments for weights of connections. Then the validation data sets are used to confirm the networks generalization ability. Validation data sets in this case are 7 samples from the first member, 5 samples from the second member, 7 samples from the third member and 6 samples from the fourth member. Quality of the classifier is evaluated using the test data set. The test data set used in this case includes 8 samples from the first member, 8 samples from the second member, 9 samples from the third member and 9 samples from the fourth member.

Mean squared error (MSE) and percentage of errors (%E) obtained for a particular iteration is indicated in the following figure 5.4 :



	Samples	MSE	%E
Training:	109	1.75897e-7	0
Validation:	25	1.82664e-7	0
Testing:	34	1.80729e-7	0

Figure 5.4: Representation of MSE and %E

As we all know, mean squared error (MSE) indicates the average squared difference between outputs and targets. A low value of MSE indicates better results and zero means there are no errors.

Performance of this test can be understood from the following graph 5.5 :

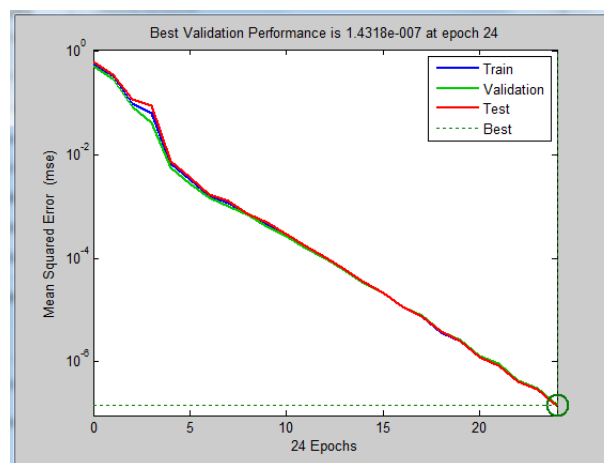


Figure 5.5: Performance plot of ANN

With MFCC feature 100% accuracy have been obtained in the speaker

recognition process. Then the same experiments were conducted for all other features and a combination of different feature sets mentioned in the tables - table 5.1 and table 5.2.

5.6 Speaker Recognition with k-Nearest Neighbor and Support Vector Machine Classifiers

5.6.1 k-NN Classifier

The k-NN classifier is a non-parametric method for classification and regression. This method predicts objects values or class memberships based on the k closest training examples in the feature space. k-NN is a type of instance-based learning or lazy learning where the function is only approximated locally and all computation is deferred until classification. k-NN algorithm is amongst the simplest of all machine learning algorithms. Using this classifier an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor [Encyclopedia 2013]. Different methods that employ k-NN for its speaker recognition activity are available in the literature [Lu, Zhang, and Jiang 2002; Lu, Jiang, and Zhang 2001].

5.6.2 SVM Classifier

In machine learning, SVMs (also known as support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Basic SVM takes a set of input data and predicts for each given input,

which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVMs are able to perform both linear and non-linear classification [Encyclopedia 2013]. Different algorithms that are proposed towards SVM speaker recognition are studied from the papers [Hatch, Kajarekar, and Stolcke 2006; Campbell et al. 2006; Solomonoff, Campbell, and Boardman 2005; Solomonoff, Quillen, and Campbell 2004; Campbell, Sturim, and Reynolds 2006].

As with the ANN classification scheme, both k-NN and SVM classifiers need a large set of feature data that are extracted in the feature extraction module from a diverse collection of recorded speech signals. Classification is conducted on different groups of data sets to test different combinations of features extracted from the speech signals using Matlab.

5.6.3 Training Data

Speaker recognition modeling with k-NN/SVM also uses features from the same speech database that are collected and thus includes $6 \times 7 \times 10$ speech samples. With the data from five males and five females speakers and with 6 speech samples of each 7 speech signals the k-NN/SVM classifier is trained so as to obtain good results with better robustness and might be able to differentiate between speakers of both sexes with similar precision.

A total of 420 initial sound samples were taken using music editor free sound recorder. First five utterances of each word from every person (350 training samples) were used as training data. Rest of the samples (70 testing samples) were used for validation and test purposes in these classification schemes.

In order to test different combinations of data and features, the data sets are also grouped in different combinations and saved as different files. A total of 28 different data sets have been created with different combinations of speakers and features are extracted from each of these samples. In order to avoid biasing of the classifier towards any speaker, the number of samples selected for training should be same for all speakers. Also an attempt is given to create the training set in a rather random manner that helps the classification scheme to be not trained towards any speaker specifically and also to train the network evenly among the speakers.

As in ANN, four different types of inputs, as shown in table 5.1, with ten speaker combinations for each, as presented in table 5.2 are used.

5.6.4 Testing Data

Once the training is done, next phase involves testing the extracted feature values with two types of data for text dependent and text independent speech recognition activities.

As the first step, network is tested with 1 or 2 utterances for each of the words or sentences used in training. Testing is done so that the classifier should be able to recognize the speaker with extreme probability for text-dependent data sets. Performance of the same scheme is also tested with text-independent data sets. Here what we have done is to select a set of test data that are not present in the train data sets. Comparison is performed

on the text-dependent and text-independent data for single or different combinations of features.

5.6.5 Experimental Results

In this case, the same data sets as in the ANN classification scheme are used. As in the above case, the voice database consists of 10 speakers with 5 male and 5 female speakers in the age group of 28 - 38. Speech samples were recorded with a sampling frequency of 44,100 Hz with 16-bits per sample with the help of music editor sound recorder.

Feature Sets

Feature sets are created by first grouping the same feature values from different speech samples and then consider different combinations of these feature sets. The first feature set includes MFCC coefficients of 420 speech samples. Likewise testing is conducted independently with individual feature sets for all the selected samples, then consider different combinations of these feature sets towards the training and testing data sets. Thus the first group of feature sets consisted of 20 MFCC coefficients or other feature values listed in table 5.1. The second group of feature sets includes a combination of 2 different data sets or a combination of 3 different data sets and so on of the same table 5.1. The feature sets selected for training as well as testing schemes varies depending upon the type or types of features selected in the process of classification.

The classification scheme performed in this work employs two different speaker recognition schemes such as speaker verification and speaker identification. Speaker verification approach tests unknown samples with the train data sets. On the other hand, the speaker identification approach

takes only known samples with the train data sets. Another scheme employed here includes a combination of unknown and known samples to test with a set of train data. Of these three schemes, the third one matches more closely to the real world scenario.

Speaker Verification Approach

Following figures 5.6, 5.8 & 5.10 show the distribution of test data sets on the training data sets for k-NN and figures 5.7, 5.9 & 5.11 for SVM classifiers respectively. In speaker verification unknown samples are taken as the test data sets and the result obtained reveals the distribution of these data sets on the data sets available for the same speaker in the train data sets. In below figures, blue circles corresponds to the test data set, green circles represents the correctly classified items and red circles represents the incorrectly classified items on each of the classes and it reveals that using the MFCC values, speaker verification can be achieved to a good extent.

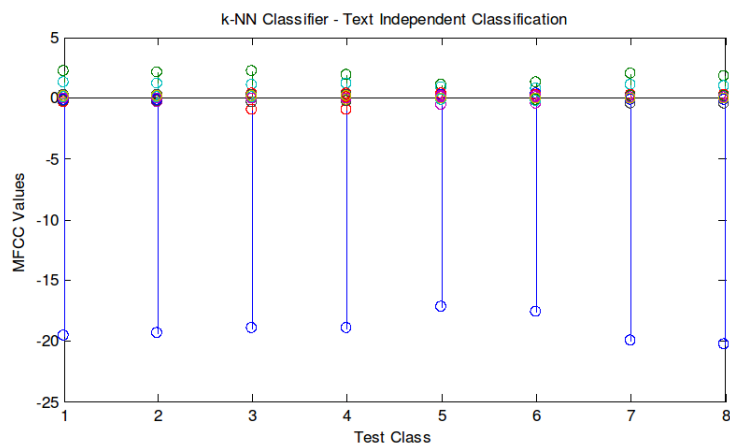


Figure 5.6: k-NN speaker verification



Figure 5.7: SVM speaker verification

Speaker Identification Approach

Speaker identification is the second approach in speaker recognition task. Here a known set of data have been taken for in classification and obtained good results on execution of the code. In these figures also blue circles corresponds to the test data set, green circles represents the correctly classified items and red circles represents the incorrectly classified items on each of the classes. Speaker identification can be achieved to a good extent by using the MFCC values.

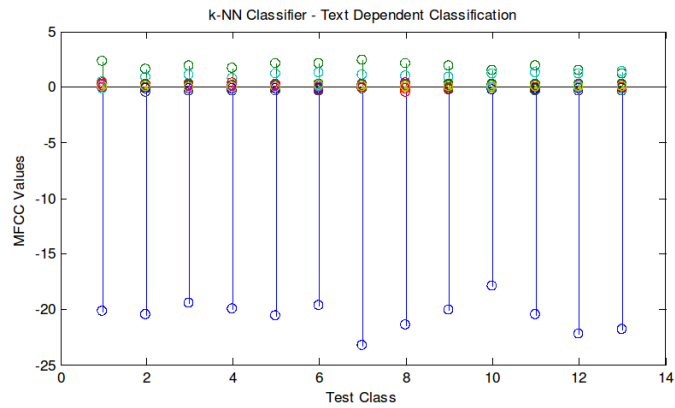


Figure 5.8: k-NN speaker identification

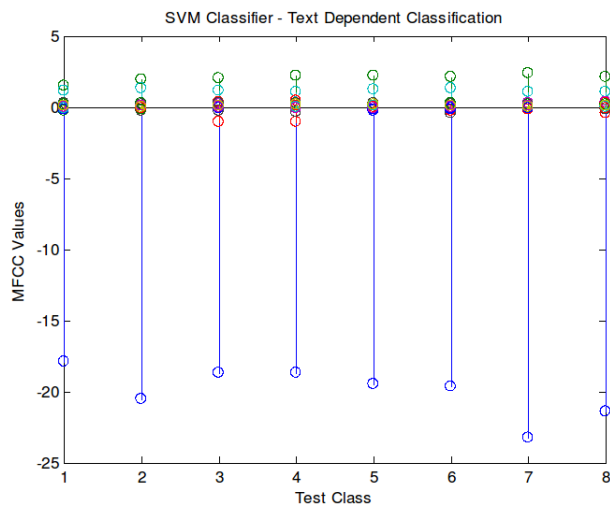


Figure 5.9: SVM speaker identification

Speaker Recognition Approach

In the third scheme a known set as well as an unknown set of samples have been employed against the train data sets. Intention behind this task is to make the work more realistic with data set that matches more closely to the real world scenario. Here also, blue circles corresponds to the test data sets, green circles represents the correctly classified items and red circles represents the incorrectly classified items on each of the classes and speaker recognition can be achieved to a good extent.

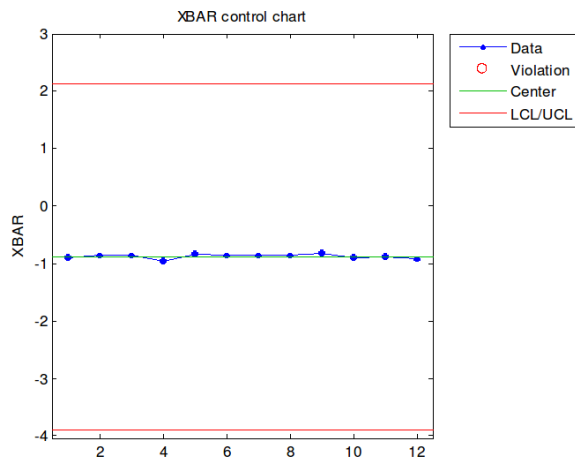


Figure 5.10: k-NN speaker recognition

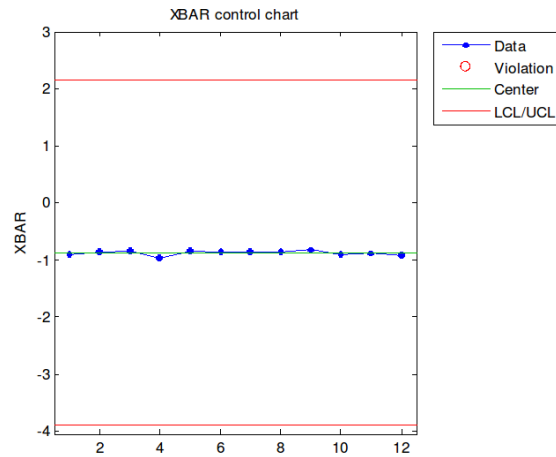


Figure 5.11: SVM speaker recognition

Thus it can be concluded that both k-NN and SVM classifiers also give better results in the speaker recognition task for a known and unknown data set or for the combination of these two data sets.

5.7 Comparative Study of ANN, k-NN and SVM

This section deals with a comparative analysis of the speaker recognition results obtained on different features as well as on its different combinations. Following table 5.3 demonstrates the features and its combinations that are employed in the classification schemes.

Table 5.3: Feature selection

Signal Features	MFCC	SF	SR	SC	ZCR	EE	SE
MFCC	1	1	1	1	1	1	1
SF	0	1	1	1	1	1	1
SR	0	0	1	1	1	1	1
SC	0	0	0	1	1	1	1
ZCR	0	0	0	0	1	1	1
EE	0	0	0	0	0	1	1
SE	0	0	0	0	0	0	1

Below tables 5.4, 5.5 and 5.6 indicate the percentage of correctly classified speakers for the ANN, k-NN and SVM classifiers.

Table 5.4: Classification accuracy for single features

Classifiers	MFCC	SC	SR	SF	ZCR	EE	SE
ANN	100%	95.1%	96.33%	97.5%	97.5%	89.5%	90.5%
kNN	100%	90.11%	85%	78.25%	77.25%	60.75%	64.25%
SVM	100%	91.23%	85%	79.33%	78.33%	61.23%	63.23%

Speaker recognition module is performed for individual signal features as well as for a combination of these features. This module enables us in identifying the signal features that help in speaker recognition tasks to a great extent. Features or feature combination identified in the process are employed towards the development of data codes which are treated as watermark in the proposed watermarking schemes.

Table 5.5: Classification accuracy for a combination of features

Classifiers	All Features (including MFCC)	SR & SF	SR & ZCR	SR & SC	SR & EE
ANN	98.1%	88.50%	90.90%	90.90%	89.50%
kNN	89%	85.00%	82.50%	79.25%	76.50%
SVM	91.54%	80.00%	73.50%	76.92%	76.15%

Table 5.6: Classification accuracy for a combination of features

Classifiers	SF & EE	SF & SR & ZCR	SC & SR & ZCR	SC & SE & SR
ANN	87.50%	87.50%	98.10%	90.90%
KNN	83.25%	87.11%	86.92%	86.25%
SVM	83.84%	88.54%	81.54%	83.85%

From the tables 5.4, 5.5 and 5.6 , it can be concluded that the MFCC feature itself can be utilized towards the speaker recognition activity.

5.8 Summary

The speaker recognition or the classification module determines the best features that are employed towards the generation of signal dependent watermark. Features are selected in such a way that, they can by itself or with a combination of other feature sets have the ability to distinguish the speakers reliably. Initially, individual features are considered and then a combination of these features with different classifiers such as ANN, k-NN

and SVM. Thus the experimental results reveals that the classification module helps in the selection of ideal and most reliable characteristics (from the features that are extracted) for speaker recognition towards the preparation of the FeatureMark.

Chapter 6

Barcode Based FeatureMarking Scheme

6.1 Introduction

Aimed at multimedia data, a wide range of effective watermarking algorithms have been proposed and implemented. As far the audio watermarking schemes are concerned, the number of algorithms suggested is very less compared to other multimedia watermarking schemes. The reason behind this is that, the human auditory system (HAS) is far more complex and sensitive than the human visual system (HVS). With electronic communications in this digital era various kinds of disputes may arise in digital audio communication and these disputes may be the denial of authorship of the speech signal, denial of sending or receiving the signal, denial of the time of occurrence etc. The significance of a non-repudiation service arises in these circumstances which guarantee the evidence of an occurrence of a particular event, the time of occurrence and the integrity of the parties

involved.

Developing a non-repudiate voice authentication scheme is a challenging task in the context of audio watermarking and this research aim to suggest a digital audio watermarking scheme that ensures authorized and legal use of digital communication, copyright protection, copy protection etc. that helps to prevent such disputes. This chapter introduces a voice signal authentication scheme that employs the FFT towards the embedding and detection schemes and a signal dependent feature for its watermark generation.

6.2 Fourier Analysis

Acoustic signals that are time domain in nature need to get converted into transform domain for retrieving the spectral information. Transforming a signal implies converting time domain information into its frequency domain which demonstrates the details of amplitude and phase components that constitute the signal. Furthermore, there exists an inverse Fourier transform that helps to retrieve the original time domain from its complex frequency domain. The conversion does not result in any loss of its information. It can be concluded that, both time domain and transform domain data are equivalent with different view for a given signal. FFT is computationally efficient method for evaluating the Fourier transform of a digital (acoustic) signal. In order to work with this, the time domain signals must be segmented into finite length blocks of sampled data called frames. Then length of FFT frames is evaluated in terms of sampling rate. In units of time, the duration on which each FFT frame observes the continuous input signal is evaluated. As the duration increases, window size, time between

two consecutive FFT spectrums and FFT data also increase [Bastani and Behbahani 2011; Encyclopedia 2013; Mathworks 1984].

Following figures [figure 6.1, figure 6.2] show the time domain representation of a voice signal plotted using praat5350 win64 tool with a sampling rate of 44100.

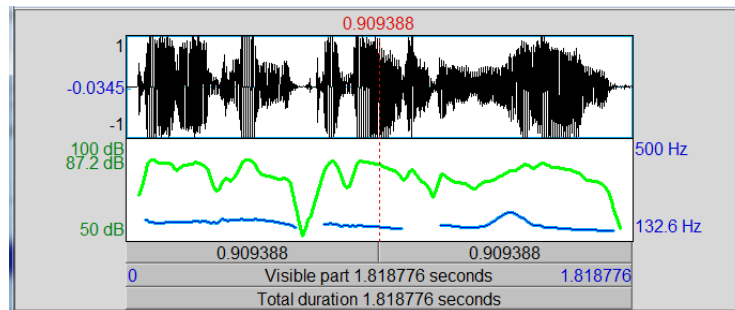


Figure 6.1: Amplitude-time plot

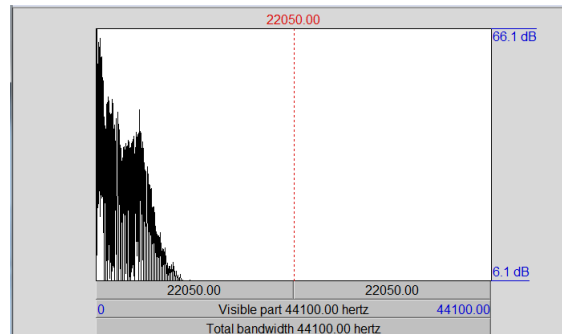


Figure 6.2: Spectrum of the signal

Above spectrum information reveals the following facts:

- Frequency domain:

- Lowest frequency: 0 Hz
- Highest frequency: 44100 Hz
- Total bandwidth: 44100 Hz

- Frequency sampling:
 - Number of frequency bands (bins): 65537
 - Frequency step (bin width): 0.336456298828125 Hz
 - First frequency band around (bin center at): 0 Hz

- Total energy: 0.18738784 Pa^2 sec

The signal sampled at 1000Hz can be plotted as figure 6.3 :

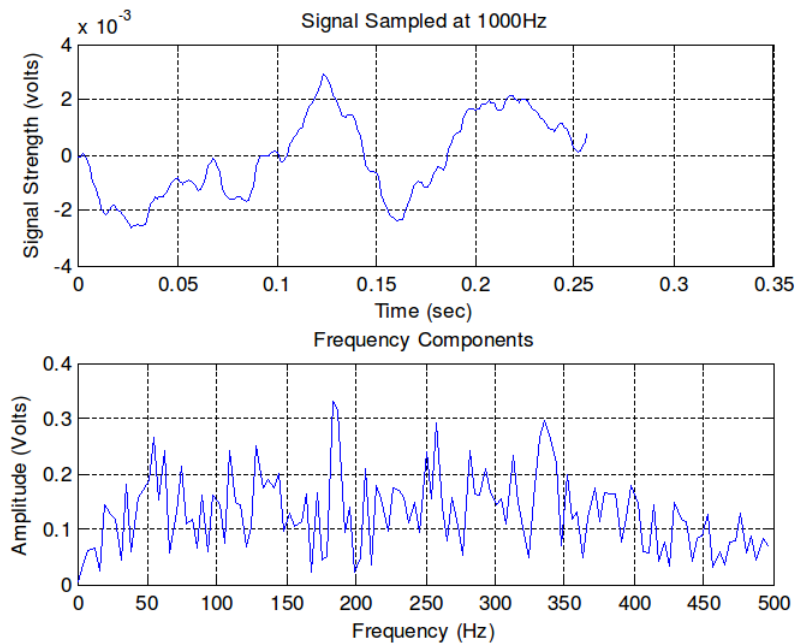


Figure 6.3: Amplitude and frequency plots

Frequency domain data generated with FFT is complex and is represented in terms of amplitude and phase and linearly spaced in terms of frequency. Time constant which evaluates the length of FFT frames in units of time is inversely proportional to its frequency resolution. FFT spectrum displays the complex frequency data spaced on uniform intervals of its frequency resolution. From this, it is easy to identify the inverse relation between time and frequency domains. Longer FFT size offers higher resolution on spectral data but with slow time response and shorter FFT size offers lower spectral resolution but with faster time response.

Analysis of FFT reveals that the frequency domain spectral data is distributed on a linearly spaced constant frequency interval. A graphical representation employs a frequency axis with equal frequency intervals, denoted in terms of Hz. And it is eminent that, the low-frequency resolution can be maximized by increasing the size of FFT frames but it increases the time duration and thus slower time response. Human auditory system perceives the acoustic pitch or frequency in equal frequency ratios and can be plotted by taking its logarithmic frequency in one axis. While distributing this FFT data in logarithmic frequency axis, it can be observed that the apparent distribution is not constant. Obviously, we require additional methods for viewing the FFT data in a perceptually significant manner to better correlate the graphical data with human hearing [Aeroflex 2005; Hyperphysics 2006].

Intensity of the signal Vs time is represented in the following figure 6.4

:

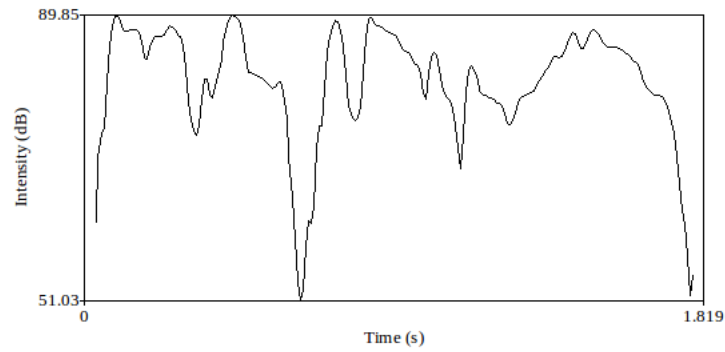


Figure 6.4: Intensity-time plot

Spectrogram representation is also shown in figure 6.5.

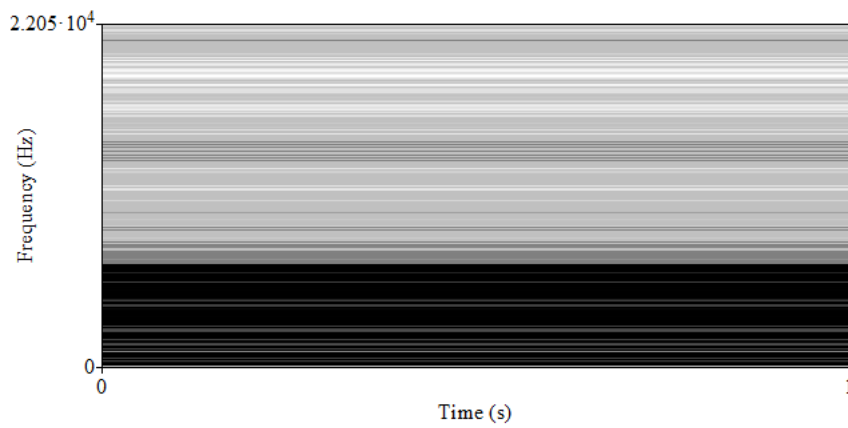


Figure 6.5: Spectrogram

Inverse Fourier transform on frequency domain data recovers the original time domain signal. IFFT is the function employed towards this process. And the 'symmetric flag' helps to nullify the numerical inaccuracies or in

other words to zero out the small imaginary components and to recover the original signal. Original time domain signal and the recovered time domain signal behave almost identically and does not reveal any significant difference while playing the audio. In an ideal or real world scenario, it might be able to evaluate the FFT of an acoustic signal for a finite duration or constant time window. And a generalization will be deduced towards the entire signal and may lead to irregularities in the frequency domain and thus to errors in the observed frequency spectrum.

The effect of Fourier transform on an infinite time sequence, on a rectangular window or on a hamming window is described in the above figure 6.5. FFT is able to accurately determine the spectrum and could identify unique steady state frequency of the signal. Infinite time domain signal can be reduced to a finite length frame by using rectangular windowing function and is observed that the energy is dispersed into a single main lobe and a number of side lobes. But the rectangular windowing results in abrupt changes towards its edges or the sidelobes and it can be reduced by applying a gentle time windowing function such as Hamming windowing which in effect reduces the level of side lobes to zero at the edges and also masks the spectral details from neighboring frequency components.

Fourier analysis can be viewed as representing input data using orthogonal basis. Basic element here is the sine function. Fourier decomposition thus represents the input data as a superposition of vibrations on the basis elements. Often, there are a few principal frequencies that account for most of the variability in original data.

6.3 One Dimensional and Two Dimensional Data Codes

Encoding data as its 1-D and 2-D form can be achieved with barcodes, data matrix codes or QR codes. Each of these representations is composed of two separate parts: *the finder pattern*, which is used by the scanner to locate the symbol and *the encoded data* itself. The finder pattern defines shape (square or rectangle), size, X-dimension and number of rows and columns in the symbol. The data is then encoded in a matrix within the finder pattern. A data carrier represents data in a machine readable form used to enable automatic reading of the element strings [Scandit 2011; Reinhardt 2011; Encyclopedia 2013]. In this scheme, the data carrier employed is barcode and is discussed below.

A barcode is an optical machine readable representation of data relating to the object to which it is attached. Originally barcodes systematically represent data by varying widths and spacings of parallel lines and may be referred to as linear or one-dimensional (1D). Later they evolved into rectangles, dots, hexagons and other geometric patterns in two dimensions (2D). Although 2D systems use a variety of symbols, they are generally referred to as barcodes as well. Barcodes originally were scanned by special optical scanners called barcode readers. Later, scanners and interpretive software became available on devices including desktop printers and smartphones.

6.4 Proposed Scheme

As described in Chapter 4, original speech signal is divided into frames with length l . Applying FFT on these frames generates its magnitude spectrum.

The magnitude values obtained are arranged in a matrix which is then transformed into its frequency domain in order to embed the watermark bits.

Let $V = v(i), 0 \leq i < Length$ represent a host digital audio signal with $Length$ samples. $FM = FM(i, j), 0 \leq i < M, 0 \leq j < N$ is a binary image to be embedded within the host audio signal and $FM(i, j) \in 0, 1$ is the pixel value at (i, j) .

6.4.1 Watermark Preparation

Feature extraction module as stated in chapter 4, extracts the signal dependent physical features including the MFCC. This scheme employs MFCC values in the generation of barcode which will be treated as its signal dependent watermark.

Quantized MFCC values are streamed in desired format and submit as input to an online barcode generator. Obtained barcode holds all quantized feature vectors of the recorded voice signal. An example of the generated barcode is shown in figure 6.6. Watermark in the proposed scheme is also stated as the FeatureMark.



Figure 6.6: Sample barcode

6.4.2 FeatureMark Embedding

Obtained FeatureMark bits are first scrambled using Arnold transform which dissipate pixel space relationship of the FeatureMark image and thus helps to improve robustness criteria of the watermarked signal. Scrambled

FeatureMark image will then be converted into a 1-dimensional sequence of 1s and 0s (binary digits).

Let $FM = FM(i, j), 0 \leq i \leq M, 0 \leq j \leq N$ represents the original FeatureMark image.

Applying Arnold transform results in a scrambled structure which can be represented as figure 6.7

$FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N.$

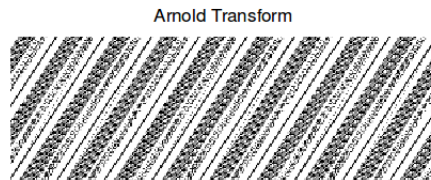


Figure 6.7: Arnold transformed barcode

Number of iterations selected to generate the Arnold transform is 2. Scrambled structure is then converted into a sequence of 1s and 0s as follows:

$FM2 = fm2(k) = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N,$

$k = i \times N + j, fm2(k) \in 0, 1$ Finally, each bit of the FeatureMark data is mapped into the signal frames by transforming it using FFT. Embedding scheme employs the following condition towards mapping of each FeatureMark bits.

Let $f_1, f_2 \dots f_n$ be the Fourier coefficients obtained. These values are sorted in an ascending order, let it be $f'_1, f'_2 \dots f'_n$ and take first 8 peak coefficients such as $f'_n, f'_{n-1} \dots f'_{n-7}$ towards FeatureMark embedding. Thus, each frame can hold 8 FeatureMark bits.

FeatureMark Embedding Scheme (represented in figure 6.8) is presented in the following algorithm:

Algorithm 1 FeatureMark embedding

Inputs

- 1: Original speech signal
- 2: FeatureMark

Output

- 1: FeatureMarked speech signal

Steps

- 1: Prepare the barcode as FeatureMark using an online barcode generator
 - 2: Generated FeatureMark is a two-dimensional sequence represented as:
 - $FM = FM(i, j), 0 \leq i < M, 0 \leq j < N$ where $FM(i, j) \in 0, 1$ is the pixel value at (i, j)
 - 3: Arnold transform is applied to scramble the FeatureMark image denoted as:
 - $FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N$
 - 4: Scrambled structure will then be converted into a one-dimensional sequence of 1s and 0s represented as:
 - $FM2 = fm2(k) = f1(i, j), 0 \leq i \leq M, 0 \leq j \leq N, k = i \times N + j, fm2(k) \in 0, 1$
 - 5: Convert the speech signal into speech samples with a sampling rate of 44100
 - 6: Apply FFT to the original speech signal and the FFT frames are segmented into non-overlapping subsegments
-

Sampling rate in context of signal processing refers to the average number of samples per second to represent the event digitally. In most digital audio, common sampling rate/frequency is 44,100 Hz as it allows for a 2.05 kHz transition band which is used in compact discs. Sampling rate

44,100 Hz was originated in the late 1970s with PCM adaptors where digital audio was recorded on video cassettes (the Sony PCM-1600 (1979) and subsequent models in this series). Nyquist-Shannon sampling theorem suggest ideal sampling rate to be greater than twice the maximum frequency one wishes to reproduce. As hearing range of human ears is roughly 20 Hz to 20,000 Hz the ideal sampling rate therefore had to be greater than 40 kHz [Encyclopedia 2013].

Algorithm 1 Featuremark embedding (continued)

7: Do for each subsegment

- Calculate the magnitude and phase spectrum of each subsegment using the FFT : Let $f_1, f_2 \dots f_n$ be the magnitude coefficients obtained
- Sort the magnitude coefficients in ascending order of their values: Let it be $f'_1, f'_2 \dots f'_n$
- Energy entropy of each frame is then calculated using the following equation 6.1

$$I_j = - \sum_{i=1 \dots k} \sigma_i^2 \log_2 \sigma_i^2 \quad (6.1)$$

- Obtain the first 8 peaks of these magnitude coefficients, let it be $f'_n, f'_{n-1} \dots f'_{n-7}$
- Insert watermark bits into these coefficients by using the following condition:
 1. For embedding a 1 to f_n , make $f_n'' = f_n' + \alpha$
 2. For embedding a 0 to f_n , make $f_n'' = f_n' - \alpha$, where $\alpha = 0.0001$
 3. Repeat embedding
- Perform the same sequence on the subsequent segments till all the watermark bits are embedded

8: Inverse FFT is applied to convert frequency domain back to original time domain to form the FeatureMarked speech signal

Here, f_i is a magnitude coefficient into which a watermark is embedded, f_{m_i} is a watermark bit to be inserted into f_i , α is a scaling factor, f_i'' is an adjusted magnitude coefficient. Energy entropy of each frame is evaluated to confirm that the embedding does not destroy energy distribution of the signal.

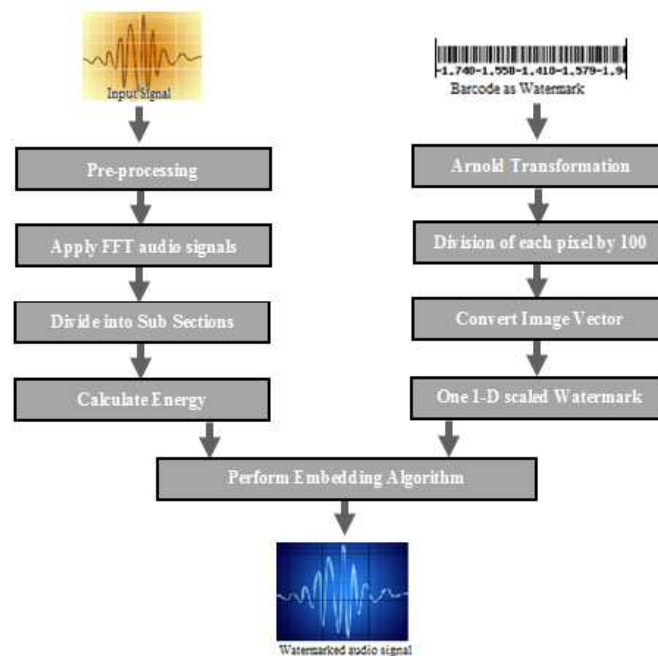


Figure 6.8: Watermark embedding scheme

6.4.3 Repeat Embedding

Embedding a single FeatureMark for an entire acoustic signal does not guarantee any robustness against common signal manipulations as well as desynchronization attacks. Excellent robustness is a key criterion of every

watermarking scheme. Robustness against these attacks can be improved by embedding the FeatureMark more than once in a particular signal. The term coined to demonstrate it is the ‘repeat embedding’ and is employed to reduce the number of ‘bit errors’ aroused due to these attacks. Let ‘ r ’ denote the number of repetitions and FM_l denote the number of bits of the FeatureMark that need to be embedded in the signal. For example, if r is 1, only FM_l bits are embedded and if r is 5, five times FM_l bits are embedded. In other words, a total of $5 \times FM_l$ bits are embedded. That is, each FeatureMark bit sequence is repeatedly embedded r times in an acoustic signal. The average number of repetitions used in our case is three. Thus, the precision or accuracy of the extracted FeatureMark can be improved to a great extent.

Let the generated watermark bits holds 2000 bits for signal duration of 1 minute then theoretical evaluation reveals the need for 5 seconds to embed the watermark and gives a prospect to embed the watermark 12 times. From the above equation deriving capacity, actual capacity of the proposed watermarking scheme is 33.33 bps.

6.4.4 Signal Reconstruction

After embedding r times the FeatureMark bits on the acoustic signal, an inverse FFT is applied to produce the speech signal. That is, modified spectra of the time domain signal is firstly converted using inverse FFT. Then, combining each of the non-overlapping or overlapping window-frame series respectively helps to reconstruct the FeatureMarked time domain signal.

Signal reconstruction can be described as follows:

- Embedding process preceded by the computation of non-overlapping Hamming windowed FFT. The original speech signal denoted as $v_1(n)$ and the results obtained is a spectra denoted as $V_1(\omega)$.
- Obtained spectra $V_1(\omega)$ is then manipulated to embed the FeatureMark bits which in turn results in the modified spectra denoted as $V_1'(\omega)$.
- At the time of speech signal reconstruction convert the $V_1'(\omega)$ to $v_1'(n)$ using inverse FFT. Then combine $v_1'(n)$, the non-overlapping Hamming window $w_1(n)$ and the overlapping Hamming window $w_2(n)$ together to reconstruct the original speech domain signal.

From this, it is understood that FFT transform on the overlap regions is not ideal for embedding the FeatureMark bits since it may be thrown away during signal reconstruction procedure. It can be summarized as follows: Let $fft(m)$ provides spectrum of the frame 'm' which is modified for embedding the FeatureMark bits; its subsequent frame 'm+1' is also get modified to satisfy the embedding criteria because m and m+1 are overlapping frames. That is, spectral modification applied to a particular frame will get distorted by spectral modifications applied to its subsequent overlapping frame. This will result in reduced accuracy of the extracted FeatureMark bits as the modification on each frame represents the embedded FeatureMark bits. In short, it can be stated that, for an FFT based FeatureMarking scheme the spectrum of each frame of the original speech signal is computed using FFT and the magnitudes of this spectrum is arranged into a matrix of order $M \times N$.

6.4.5 Watermark Detection

FeatureMark detection procedure employed in this scheme does not need the original speech signal but it needs the length or number of embedded mark bits. The detection process mainly involves identification of the presence of embedded mark bits.

In order to detect the FeatureMark bits, spectrum of each frame of the FeatureMarked signal is evaluated and the obtained magnitudes are arranged into matrix of order $M' \times N'$.

6.4.6 Digital Watermark Extraction

Once the presence of the FeatureMark is detected, next step is to extract (shown in figure 6.9) the FeatureMark bits by employing the conditions mentioned in the embedding module.

But in this scheme, entire signal should be traversed to identify the presence of FeatureMark as well to extract the mark bits. According to this condition, it extracts the FeatureMark bits and the number of bits extracted should match with that of the embedded FeatureMark bits.

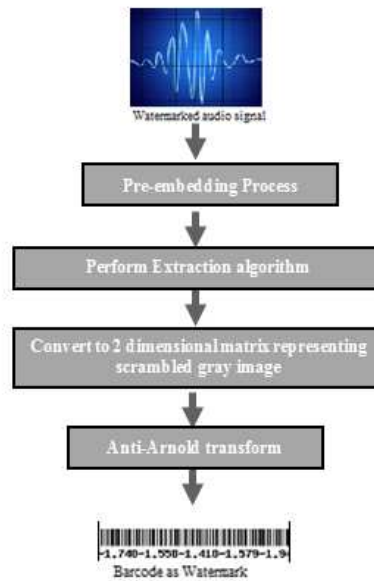


Figure 6.9: Watermark extraction scheme

Algorithm 2 FeatureMark extraction

Inputs

- 1: FeatureMarked speech signal

Output

- 1: FeatureMark

Steps

- 1: The FeatureMarked speech signal is transformed into FFT domain.
-

Algorithm 2 FeatureMark extraction (continued)

2: Do for each subsegment

- Calculate the magnitude and phase spectrum of each subsegment using the FFT :
- Let f_1, f_2, \dots, f_n be the magnitude coefficients obtained
- Sort the magnitude coefficients in ascending order of their values: Let it be f'_1, f'_2, \dots, f'_n
- Energy entropy of each frame is then calculated using the following equation 6.2

$$I_j = - \sum_{i=1 \dots k} \sigma_i^2 \log_2 \sigma_i^2 \quad (6.2)$$

- Obtain the first 8 peaks of these magnitude coefficients, let it be $f'_{n-7}, \dots, f'_{n-2}, f'_{n-1}, f'_n$
- Extract the watermark bits from these coefficients by using the following condition:
 1. Extract a '1' for $f''_n = f'_n + \alpha$
 2. Extract a '0' $f''_n = f'_n - \alpha$, where $\alpha = 0.0001$
 3. Perform the same sequence on the subsequent segments till all the watermark bits are extracted

3: Convert the one-dimensional sequence into a two-dimensional scrambled structure

4: Apply the anti-Arnold transformation on this scrambled structure to obtain the original FeatureMark image

5: Repeat Steps 2, 3 and 4 upto r times

6: Inverse FFT is applied to convert the frequency domain back to its time domain (if needed)

6.5 Experimental Results

In order to carry out the watermarking as well as to evaluate the performance of this scheme, various tests were conducted. These tests include imperceptibility tests, robustness tests and capacity tests. Experimental set up for this proposed scheme is illustrated as follows: 50 audio signals of around 10 members with a sampling rate of 44,100 times per second. The Matlab version R2009b is used in embedding and detection schemes. Music editor sound recorder is employed for inducing some of the common signal manipulations and desynchronization attacks in the signal.

Audio signals in the test are based on Malayalam speech signals with 16 bits/sample, 44.1 kHz sample rates. Signal duration vary for each signal and is in the range of 2 - 300s. Number of frames employed towards signal processing depends on the length of the signal with a frame rate of 100. From the experiments conducted it is obvious that the embedding scheme works fine with around 50 samples. Prepared FeatureMarks: Barcodes (figures 6.10 & 6.11)



Figure 6.10: Sample 1



Figure 6.11: Sample 2

Experiments were conducted on both single channel and multi-channel speech signals and the results are plotted in following figures - figure 6.12 & figure 6.13:

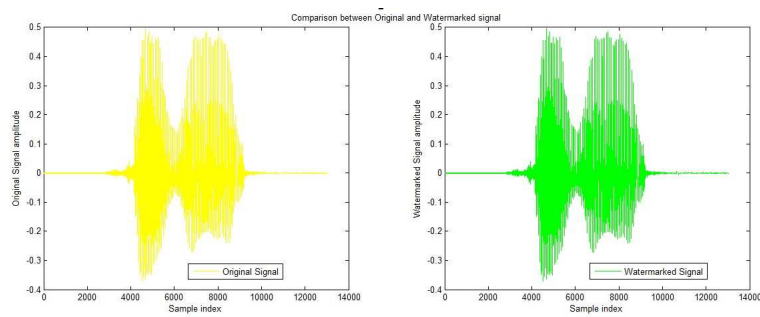


Figure 6.12: Single channel - original and FeatureMarked speech signal

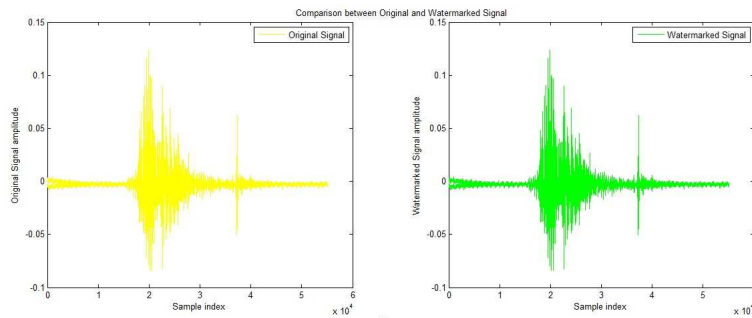


Figure 6.13: Multi-channel - original and FeatureMarked speech signal

Transparency Tests

Transparency of this proposed method is evaluated by employing the subjective listening tests. Selection of listening panel is based on a set of expert as well as non-expert listeners. In this scheme a set of non-expert listeners and used and the panel includes representative of the general population - fellow research scholars and colleagues. The final decision is based on a 5-point grade scale [ITU-R 2002] presented in table 6.1.

Table 6.1: 5-Point grade scale

Quality	Impairment
5 Excellent :	5 Imperceptible
4 Good :	4 Perceptible, but not annoying
3 Fair :	3 Slightly annoying
2 Poor :	2 Annoying
1 Bad :	1 Very annoying

And the result obtained for this scheme is given below:

Table 6.2: Imperceptibility criteria

Algorithm	Imperceptibility
FFT & Barcode	Good

Robustness Tests

Robustness of this scheme is evaluated by performing common signal manipulations or desynchronization attacks on the FeatureMarked signals.

- Measure of strength of watermarking scheme against common signal processing functions is described below:

Table 6.3: Common signal manipulations

Noise Addition:	Added white Gaussian noise with an SNR of 50
Silence Addition:	A silence of 100 ms duration is inserted at the beginning of the FeatureMarked signal
Echo Addition:	Added an echo with a delay of 200 ms
Re-sampling:	Down sampled to frequencies 22.05 kHz and then up sampled to its original 44.1 kHz
Re-quantization:	Signal with 16-bit quantized to 8-bit and then back to its original 16-bit
Low-pass Filtering:	Done with cut off frequencies 10 kHz and 200 Hz
Band-pass Filtering:	Done with cut off frequencies 10 kHz and 200 Hz

The procedure used for these common signal manipulations such as silence addition, echo addition, low-pass and band-pass filtering are performed using the freely available music editor tool. Other signal manipulations are performed in Matlab.

- Measure of strength of watermarking scheme against desynchronization attacks is shown below:

Table 6.4: Desynchronization attacks

Amplitude Variation:	Signal is amplified to its double as well as half
Pitch Shifting:	Involve frequency fluctuation to the signal
Cropping:	Performed randomly at different positions of the signal
Time-scale Modification:	Watermarked voice signal was lengthened (slow-down) and shortened (double speed)

The procedure used for desynchronization attacks such as amplification to double volume and half volume, pitch change, speed changes and random cropping are done using the Music editor software.

Performance of the proposed FeatureMarking scheme is analyzed by evaluating bit error rate (BER) of the extracted FeatureMark to the actual or embedded FeatureMark. BER is a unitless performance measure which is often expressed in %. Evaluation is performed on both single channel and multi-channel audio signals. In case of multi-channel signals, both channels are not employed for embedding the watermark bits. Instead, the two channels are added up so as to compromise the properties of the original audio signal. Results obtained are as follows.

Experiment # 1: Performed on single channel sounds with around 27 signals and was found successful on all the signals.

Table 6.5: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 1	0.0000	0.2000	0.1000	0.4200
Host Audio 2	0.0000	0.1000	0.0000	0.0000
Host Audio 3	0.0000	0.1100	0.1000	0.2100
Host Audio 4	0.0000	0.1100	0.2000	0.4000

Table 6.6: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 1	0.4200	0.2100	0.3200	0.2100
Host Audio 2	0.4500	0.1100	0.4100	0.1500
Host Audio 3	0.4800	0.1000	0.1200	0.1300
Host Audio 4	0.4000	0.4100	0.2200	0.2500

Table 6.7: Robustness test for desynchronization attacks (in BER \times 100%)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 1	0.4000	0.3900	0.2500	0.4200
Host Audio 2	0.3900	0.3800	0.3600	0.3700
Host Audio 3	0.4100	0.4100	0.4000	0.3900
Host Audio 4	0.4200	0.4200	0.3900	0.3500

Experiment # 2: Performed on multi-channel sounds with around 23 signals and was found successful on all the signals.

Table 6.8: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 5	0.0000	0.1100	0.1200	0.4200
Host Audio 6	0.0000	0.1100	0.1100	0.4000
Host Audio 7	0.0000	0.2100	0.1100	0.4000
Host Audio 8	0.0000	0.2100	0.1200	0.4100

Table 6.9: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 5	0.4100	0.2100	0.3800	0.4000
Host Audio 6	0.4200	0.2200	0.4100	0.3500
Host Audio 7	0.3700	0.2100	0.3700	0.3600
Host Audio 8	0.3800	0.4000	0.3800	0.4500

Table 6.10: Robustness test for desynchronization attacks (in BER \times 100%)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 5	0.4100	0.4000	0.3600	0.3700
Host Audio 6	0.3900	0.3900	0.4100	0.3600
Host Audio 7	0.4200	0.4100	0.3900	0.4100
Host Audio 8	0.3900	0.4100	0.3800	0.4200

Watermark recovery rate has been evaluated using the equation $(1 - BER) \times 100\%$. And the above results reveal that watermarked signals are vulnerable to common signal manipulations and desynchronization attacks.

An enhanced scheme that can withstand the common signal processing and desynchronization attacks is suggested in the upcoming chapter.

Average of the recovery rate obtained for some signals are plotted below - figure 6.14 :

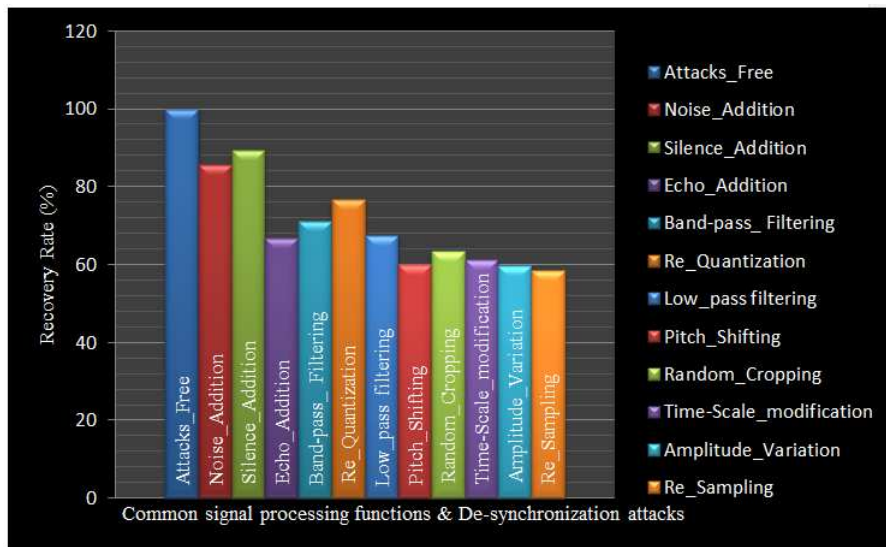


Figure 6.14: Average recovery rate

Capacity Tests

Capacity or data payload demonstrates the amount of information that can be embedded and recovered in the audio stream. It can be evaluated using the equation 6.3.

$$C = \frac{M}{L} bps, \quad (6.3)$$

where M refers to the number of watermark bits and L the length of the audio signal.

Theoretical evaluation demonstrates that this scheme holds 400 watermark bits per second because we are selecting 8 coefficients from each of the frames and a total of 50 frames have been obtained for a second. But the actual calculation provides a capacity of 66.67 bps.

6.5.1 Non-Repudiation Services

Experimental results reveal that objective of the proposed scheme, non-repudiation service [Onieva, Zhou, and Lopez 2004; Zhou and Lam 1999; Zhou and Gollmann 1997b; Zhou and Gollmann 1996a; Coffey and Saidha 1996; Onieva et al. 2004; Steinebach et al. 2001; Zhou and Gollmann 1997a; Zhou and Gollmann 1996b; Schneider 1998] could be achieved to a great extent. Realization of voice authentication scheme that guarantees non-repudiation is done by the following sequence of steps: First a signal dependent FeatureMark is developed using the physical features of signal. Then the FeatureMark is embedded by utilizing suitable FFT coefficients by transforming the signal to its frequency domain spectra. After embedding the FeatureMark bits, inverse transform IFFT is applied to convert the speech signals back to its time domain.

FeatureMarked signal is then transferred to the recipient and at recipient side, the signal is again transformed using FFT transform. Then for identifying the presence of watermark, entire signal is traversed from one end to the other end till a FeatureMark occurrence is detected. When the presence of the FeatureMark is detected, the bits are extracted and then combined to make the original FeatureMark. All the steps performed in the embedding module are reversed to make up the FeatureMark at the recipient side.

FeatureMark at the recipient side can be compared to the original one to confirm the authenticity as well as to guarantee non-repudiation. Feature

extraction module can also be conducted at the receiving side to create the same FeatureMark in case of any disputes.

Thus, it can be assured that the proposed scheme helps in achieving an authentic communication scheme for transferring audio or speech signals without causing any ownership disputes. Important characteristics of a non-repudiation scheme such as creating and storing the evidence for sender and receiver, fairness, timeliness and confidentiality could be accomplished without any difficulty.

6.6 Summary

In this scheme, a novel watermarking technique is presented using FFT for authenticating speech signals that can guarantee non-repudiation. Experimental results reveal that this scheme can be employed in authentication or copyright protection of audio sound. Obtained results also indicate that the proposed scheme has not compromised audibility of the signal. Experimental results also guarantees tolerance to various signal processing and desynchronization attacks such as noise addition, silence addition, echo addition, filtering, compression, cropping, amplitude variation and time-scale modification. The only drawback identified in this scheme is that, in order to extract the FeatureMark bits the entire signal should be traversed from one end to the other end which eventually increase the time it takes to detect the presence of the mark.

Chapter 7

Data Matrix Based FeatureMarking Scheme

7.1 Introduction

This chapter demonstrates a variation of the previous audio watermarking method that also employs FFT in the embedding scheme. An important difference associated with this scheme is the utilization of synchronization code that helps to improve the robustness nature as well as to reduce the time complexity of the watermarking method. This method is meant for improving the prior developed non-repudiate voice authentication scheme in the context of audio watermarking. Newly prepared watermark varies from the earlier one as it uses different signal properties. It also employs a varied embedding and detecting schemes for the watermarking activity.

As discussed in the previous chapter, Fourier analysis (section 6.2) helps us in understanding the spectra of the acoustic signal by converting from its time domain to the transform domain.

7.2 Data Matrix Code - A type of Data Code

As detailed in the previous chapter, encoding data as its 1-D and 2-D form can be achieved with barcodes, data matrix codes or QR Codes. In this scheme data matrix code is employed for watermarking and data carrying [Encyclopedia 2013; TEC-IT 2012; GS1 2010; Kaywa 2013a].

A data matrix code is a two-dimensional matrix representation which encodes text or numeric data. It can be represented as a square or rectangular pattern with varying arguments of black and white cells which is according to the information to be encoded. Length of encoded data depends on the number of cells in the matrix. A data matrix symbol can store up to 2335 alphanumeric characters [Encyclopedia 2013]. This representation is an ordered grid of dark and light dots ordered by a finder pattern. The finder pattern is partly used to specify the orientation and structure of the symbol. The data is encoded using a series of dark or light dots based upon a pre-determined size. The minimum size of these dots is known as the X-dimension. Data matrix is capable of encoding variable length data. Therefore, size of the resulting symbol varies according to the amount of data encoded. This data matrix is employed as watermark in the algorithm that is suggested in this chapter.

7.3 Proposed Scheme

This scheme perform a frame based segmentation of the original speech signal with a defined length l as mentioned in Chapter 4. Magnitude spectrum is generated by applying the FFT on these frames. In order to embed the watermark bits, frequency domain transform of these magnitude values which are arranged in a matrix form is used.

Let $V = v(i), 0 \leq i < Length$ represent a host digital audio signal with Length samples. $FM = FM(i, j), 0 \leq i < M, 0 \leq j < N$ is a binary image to be embedded within the host audio signal and $FM(i, j) \in 0, 1$ is the pixel value at (i, j) .

7.3.1 Watermark Preparation

In the generation of watermark, signal dependent physical features including MFCC, spectral roll-off and zero-cross rate obtained by the feature extraction process are employed. These feature values are used in generation of data matrix code which is the watermark in the proposed scheme.

During the data matrix code generation, quantized MFCC, spectral flux and zero-cross rate values are streamed in desired format and submitted as input to an online data matrix code generator. Obtained data matrix code (an example is shown in figure 7.1) carries all these values of recorded voice signal and can be exploited in the detection scheme. Watermark in the proposed scheme is also stated as the FeatureMark.

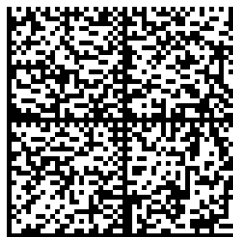


Figure 7.1: Sample data matrix code

7.3.2 Synchronization Code Generation

An important issue associated with audio watermarking schemes is synchronization [Arvelius 2003; Mathworks 1984; Encyclopedia 2013; Altalli 2008]. Watermarks are detected by aligning the watermark block with the detector. Any variation in synchronization results in false detection. Any spatial or transform domain modifications cause the detector loose its synchronization. Therefore it is advisable to use proper synchronization algorithms based on robust synchronization code. In the proposed approach, a 16-bit synchronization code is embedded in front of the FeatureMark to locate the FeatureMark position.

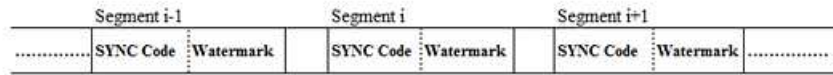


Figure 7.2: Construction of embedding information

Barker code, subsets of PN sequences, is commonly used for frame synchronization in digital communication systems. Barker codes have low correlation sidelobes. A correlation side lobe is the correlation of a code word with a time-shifted version of itself. Correlation side lobe, C_k , for a k -symbol shift of an N -bit code sequence, x_j is given by

$$C_k = \sum X_j X_{(j+k)}, \quad (7.1)$$

where X_j is an individual code symbol taking values $+1$ or -1 for $j=1,2,3, \dots, N$ and the adjacent symbols are assumed to be zero.

According to the order length of code selected for generation of Barker code any one from the following sets can be the result - figure 7.3:

Length	Barker Codes	
2	+1 -1	+1 +1
3	+1+1-1	
4	+1-1+1+1	+1-1-1-1
5	+1+1+1-1+1	
7	+1+1+1-1-1+1-1	
11	+1+1+1-1-1-1+1-1-1+1-1	
13	+1+1+1+1+1-1-1+1-1+1-1+1-1	

Figure 7.3: Barker codes

For example, for a 3^{rd} order selection, generated Barker code will be

$$\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$$

In this scheme we have generated a 13-bit

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$

Barker code using the Matlab toolkit. That is, the Barker code which we have obtained is a sequence of 1s and -1s. Next step evaluates its auto-correlation function after appending three 0s towards its end in order to make a 16-bit sequence.

And taking the integer values provides a sequence of 1s and 0s as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

which is employed as the synchronization code.

7.3.3 Embedding Method

Embedding module functions through three different levels: synchronization code embedding, FeatureMark embedding and repeat embedding (as presented in the figure 7.2).

First step associated with the embedding module is to divide the original speech signal into a set of segments and then each segment to two sub-segments. Secondly, with spatial watermarking technique, synchronization bits are embedded into the sub-segments by employing the time

domain techniques. After embedding the synchronization code, subsequent segments are transformed using FFT.

Synchronization Code Embedding

In order to guarantee robustness and transparency of watermarking, proposed scheme embeds synchronization code into the spatial domain of audio samples as follows:

Algorithm 3 Synchronization code embedding

Steps

- 1: Each audio segment is divided into sub segments with n samples
 - 2: Consider the first audio segment $A_{10}(k)$, ($k = 0, 1, 2, \dots, L$) and its subsegments
 - 3: Take four consecutive samples in this segment, let it be s_1, s_2, s_3, s_4 then
 - 4: **if** $s_1 + s_3 > s_2 + s_4$ **then**
 - 5: **if** $s_1 > s_3$ **then**
 - 6: insert 1 into s_1
 - 7: **else**
 - 8: insert 1 into s_3
 - 9: **end if**
 - 10: **else if** $s_2 + s_4 > s_1 + s_3$ **then**
 - 11: **if** $s_2 > s_4$ **then**
 - 12: insert 0 into s_2
 - 13: **else**
 - 14: insert 0 into s_4
 - 15: **end if**
 - 16: **else if** $s_1 + s_3 = s_2 + s_4$ **then**
 - 17: replace with 1 or 0
 - 18: **end if**
-

FeatureMark Embedding

For embedding the FeatureMark bits, first step is to scramble the FeatureMark image using Arnold transform which dissipate its pixel space relationship. It is performed to improve the robustness measures of the watermarked signal. Scrambled structure is also in a 2-D form which is converted into a 1-dimensional sequence of 1s and 0s and is executed as second step in the embedding task.

Let $V = v(i), 0 \leq i < Length$ represent a host digital audio signal with Length samples. $FM = fm(i, j), 0 \leq i < M, 0 \leq j < N$ is a binary image to be embedded within the host audio signal, $FM(i, j) \in 0, 1$ is the pixel value at (i, j) and $S = s(i), 0 \leq i \leq Lsyn$ is a synchronization code with Lsyn bits, where $s(i) \in 0, 1$.

Applying Arnold transform to the original FeatureMark image results in a scrambled structure which can be represented as figure 7.4.

$$FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N$$

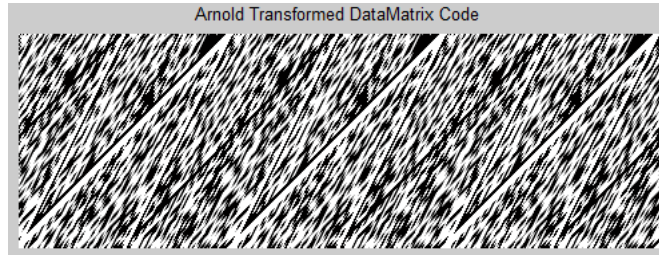


Figure 7.4: Arnold transformed data matrix code

Scrambled structure is then converted into a sequence of 1s and 0s as follows: $FM_2 = fm_2(k) = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N,$
 $k = i \times N + j, fm_2(k) \in 0, 1$

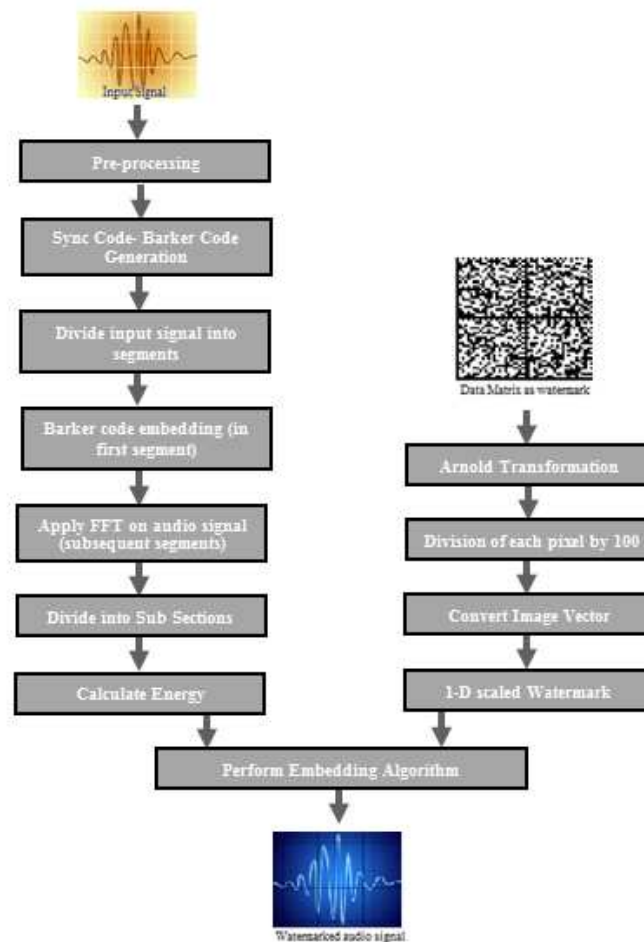


Figure 7.5: Data matrix embedding scheme

Finally, the actual embedding is performed where each bit of Feature-Mark data is mapped into the signal frames especially into the selected coefficients by transforming it using FFT. Embedding scheme (figures 7.2, 7.5) functions through the following sequence of steps:

Algorithm 4 FeatureMark embedding

Inputs

- 1: Original speech signal
- 2: FeatureMark

Output

- 1: FeatureMarked speech signal

Steps

- 1: Prepare the FeatureMark, the data matrix code using an online data matrix code generator
- 2: Generated FeatureMark is a two-dimensional sequence represented as:
 - $FM = FM(i, j), 0 \leq i < M, 0 \leq j < N$ where $FM(i, j) \in 0, 1$ is the pixel value at (i, j)
- 3: Arnold transform is applied to scramble the FeatureMark image denoted as:
 - $FM1 = FM1(i, j), 0 \leq i \leq M, 0 \leq j \leq N$
- 4: Scrambled structure will then be converted into a one-dimensional sequence of 1s and 0s represented as:
 - $FM2 = fm2(k) = f1(i, j), 0 \leq i \leq M, 0 \leq j \leq N, k = i \times N + j, fm2(k) \in 0, 1$
- 5: For the next subsegment consecutive to the SYNC embedded segment
 - Calculate the magnitude and phase spectrum of each subsegment using the FFT : Let $f_1, f_2 \dots f_n$ be the magnitude coefficients obtained
 - Sort the magnitude coefficients in ascending order of their values: Let it be $f'_1, f'_2 \dots f'_n$
 - Energy entropy of each frame is then calculated using the following equation 7.2

$$I_j = - \sum_{i=1 \dots k} \sigma_i^2 \log_2 \sigma_i^2 \quad (7.2)$$

Algorithm 4 Featuremark embedding (continued)

- 6: Take four consecutive samples in this segment, let it be f_1, f_2, f_3, f_4 then
- if** $f_1 + f_3 > f_2 + f_4$ **then**
- if** $f_1 > f_3$ **then**
- insert 1 into f_1
- else**
- insert 1 into f_3
- end if**
- else if** $f_2 + f_4 > f_1 + f_3$ **then**
- if** $f_2 > f_4$ **then**
- insert 0 into f_2
- else**
- insert 0 into f_4
- end if**
- else if** $f_1 + f_3 = f_2 + f_4$ **then**
- replace with 1 or 0
- end if**
- 7: Insert the watermark bits into these coefficients by using the following condition:
- For embedding a 1 to f'_n , make $f''_n = f'_n + \alpha$
 - For embedding a 0 to f'_n , make $f''_n = f'_n - \alpha$, where $\alpha = 0.0001$
 - Repeat embedding
- 8: Perform the same sequence on the subsequent segments till all the watermark bits are embedded
- 9: Inverse FFT is applied to convert the frequency domain back to the original time domain to form the FeatureMarked speech signal
-

where f_i is a magnitude coefficient into which a watermark is embedded, f_{m_i} is a watermark bit to be inserted into f_i , α is a scaling factor, f''_i is an adjusted magnitude coefficient. Energy entropy of each frame is evaluated to confirm that the embedding does not destroy energy distribution of the

signal.

7.3.4 Repeat Embedding

FeatureMark image is repeatedly embedded in the same acoustic signal in order to guarantee better robustness against common signal manipulations and desynchronization attacks. Thus, the watermarking scheme that have been developed should exhibit excellent robustness. An example is given in section 6.4.3. Thus, the precision or accuracy of the extracted FeatureMark is very high.

7.3.5 Signal Reconstruction

Signal reconstruction is the last step in the embedding phase. In this scheme the same steps described in section 6.4.4 are followed towards the reconstruction of the entire watermarked signal.

7.3.6 FeatureMark Detection Scheme

FeatureMark detection procedure presented in this scheme is classified under the blind watermarking techniques as it does not employ the original speech signal or any other additional information for detecting/extracting the bits. Detection scheme functions as a two-step process where the presence of synchronization code is detected first and then the FeatureMark. FeatureMark bits are extracted by keeping in mind that the bits are present between two synchronization codes.

Synchronization Code Detection

In synchronization code detection, FeatureMarked speech signal is segmented and each segment to two sub-segments. Check for the Barker code

bits in each of the sub-segments. Since the synchronization code bits are embedded in the spatial domain, presence of these bits can be directly identified in the subsegments. Following figure demonstrates this procedure - figure 7.6:

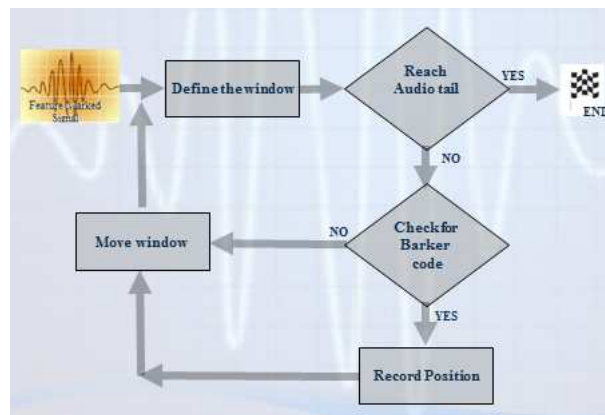


Figure 7.6: Synchronization code detection

SYNC detection process is implemented by using the steps given in Algorithm 5

FeatureMark Detection

Once the synchronization code bits are identified, FeatureMark bits are detected by evaluating the spectrum of each frame of the FeatureMarked signal. Presence of FeatureMark bits also follow another set of synchronization code bits in the subsequent segments that supports repeat embedding task. Obtained magnitudes are then arranged into a matrix of order $M' \times N'$ to finalize the FeatureMark.

Algorithm 5 Synchronization code detection

Steps

- 1: First the FeatureMarked speech signal is divided into segments and subsegments
 - 2: Consider the first audio segment $A_{10}(k)$, ($k = 0, 1, 2, \dots, L$) and its subsegments
 - 3: Check for the presence of Barker code bits in the time domain samples
 - 4: Take four consecutive samples in this segment, let it be s_1, s_2, s_3, s_4 then
 - 5: **if** $s_1 + s_3 > s_2 + s_4$ **then**
 - 6: **if** $s_1 > s_3$ **then**
 - 7: extract 1 from s_1
 - 8: **else**
 - 9: extract 1 from s_3
 - 10: **end if**
 - 11: **else if** $s_2 + s_4 > s_1 + s_3$ **then**
 - 12: **if** $s_2 > s_4$ **then**
 - 13: extract 0 from s_2
 - 14: **else**
 - 15: extract 0 from s_4
 - 16: **end if**
 - 17: **else if** $s_1 + s_3 = s_2 + s_4$ **then**
 - 18: extract the bit 1 or 0
 - 19: **end if**
 - 20: Once the bits are extracted, a comparison should be done to avoid the false positiveness
-

7.3.7 Digital Watermark Extraction

Detection of Barker code and FeatureMark are followed by the extraction module. In this scheme, presence of SYNC code reduces the burden of traversing entire signal to identify the presence as well as the extraction of FeatureMark bits.

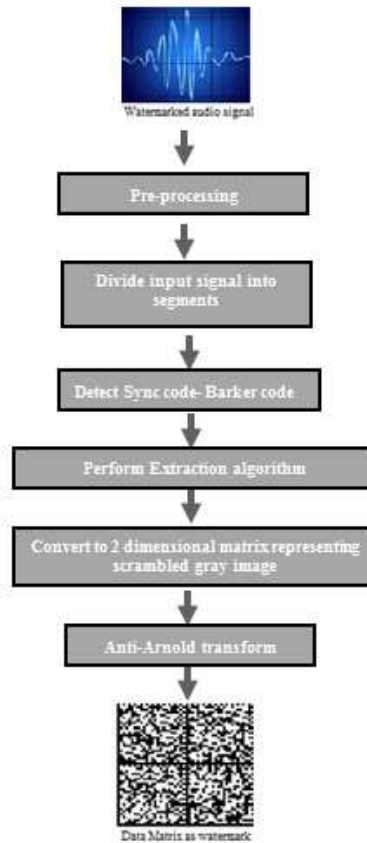


Figure 7.7: Watermark extraction scheme

Extraction of FeatureMark bits (shown in figure 7.7) also follow the

same embedding conditions employed in the embedding procedure. Extraction process is implemented by using the following steps:

Algorithm 6 FeatureMark extraction

Inputs

- 1: FeatureMarked speech signal

Output

- 1: FeatureMark

Steps

- 1: Consecutive subsegments of the FeatureMarked speech signal are transformed into FFT domain.
 - 2: Do (for each subsegment)
 - 3: Take four consecutive samples in this segment, let it be f_1, f_2, f_3, f_4 then
 - 4: **if** $f_1 + f_3 > f_2 + f_4$ **then**
 - 5: **if** $f_1 > f_3$ **then**
 - 6: extract 1 from f_1
 - 7: **else**
 - 8: extract 1 from f_3
 - 9: **end if**
 - 10: **else if** $f_2 + f_4 > f_1 + f_3$ **then**
 - 11: **if** $f_2 > f_4$ **then**
 - 12: extract 0 from f_2
 - 13: **else**
 - 14: extract 0 from f_4
 - 15: **end if**
 - 16: **else if** $f_1 + f_3 = f_2 + f_4$ **then**
 - 17: extract the bit 1 or 0
 - 18: **end if**
-

Algorithm 6 Featuremark extraction (continued)

- 19: Convert the one-dimensional sequence into a two-dimensional scrambled structure
 - 20: Apply the anti-Arnold transform on this scrambled structure to obtain the original FeatureMark image
 - 21: Inverse FFT is applied to convert the frequency domain back to the original time domain (if needed)
-

Repeat the process if it is necessary to extract one more FeatureMark from the signal.

7.4 Experimental Results

Malayalam speech signals with 16 bits/sample at 44.1 kHz sampling rate are collected and utilized in the embedding as well as the detection scheme. Performance of this scheme is evaluated by conducting various trials such as imperceptibility, robustness and capacity tests. Around 50 speech signals from 10 members were collected with a sampling rate of 44100 times per second. For implementation of this scheme Matlab R2009b version is used and common signal manipulations and desynchronization attacks are performed using the music editor sound recorder.

Employed speech signals are in the range of 2 - 300s. Data matrix code, the FeatureMark image is taken as the watermark of this proposed scheme. Number of frames employed towards signal processing depends on the length of the signal with a frame rate of 100. From the experiments conducted, it is obvious that the embedding scheme works fine with most of the samples.

Prepared FeatureMarks: Data Matrix Codes (figures 7.8, 7.9 & 7.10)



Figure 7.8: Sample 1



Figure 7.9: Sample 2



Figure 7.10: Sample 3

Single Channel Audio Signals

In case of single channel audio signals (figure 7.11), FFT coefficients are extracted directly and used for embedding the watermark bits.

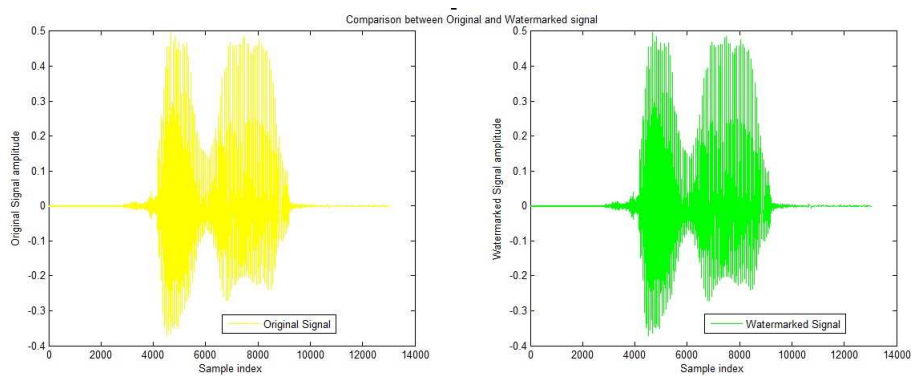


Figure 7.11: Single channel - original and FeatureMarked speech signal

Multi Channel Audio Signals

In order to work with this multi-channel or stereo signals (figure 7.12), apart from utilizing two channels separately for embedding the watermark bits, the two-channels are added together without making any changes to its spectral properties and without any perceptible deviations. For this, the two-channels have not been concatenated but the bits sequences are added together without altering its properties so that the embedding scheme functions same as for the single-channel speech signals. Both single-channel and stereo signals were recorded using the Music editor software. The work involving multiple channels can be part of future work.

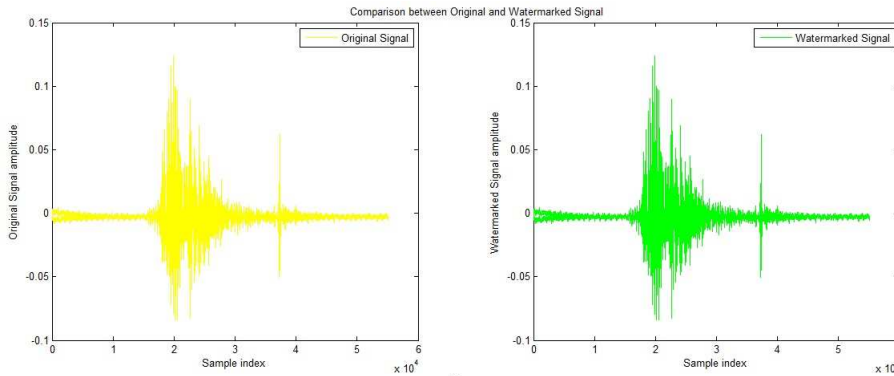


Figure 7.12: Multi-channel - original and FeatureMarked speech signal

Transparency Tests

Transparency tests evaluate effect of embedded watermark bits against the audible quality of the marked signal. In this scheme, subjective listening is employed for confirming the audio quality by a set of non-expert listeners. Due to the fact that non-expert listeners are representatives of the general population, selected people include fellow research scholars and colleagues. Final decision is based on a 5-point grade scale observed from the table 6.1.

Result obtained for this scheme is given below:

Table 7.1: Imperceptibility criteria

Algorithm	Imperceptibility
FFT & DM	Excellent

Robustness Tests

Watermark detection accuracy in this scheme is evaluated by performing a set of common signal manipulations and desynchronization attacks on the FeatureMarked signals as follows:

- Measure of strength of watermarking scheme against common signal processing functions is described in table 6.3

The procedure used for these common signal manipulations such as silence addition, echo addition, low-pass and band-pass filtering are performed using the music editor sound recorder. Other signal manipulations are performed using Matlab toolboxes.

Robustness tests on common signal processing functions give a low value for BER.

- Measure of strength of watermarking scheme against desynchronization attacks is shown in table 6.4. Desynchronization attacks such as amplification to double volume and half volume, pitch change, speed changes and random cropping are done using the music editor sound recorder.

In this scheme, performance of the proposed system is finalized by assessing BER of the extracted FeatureMark to the embedded FeatureMark. Evaluation is conducted on both single channel and multi-channel audio signals. Results obtained are shown in the following tables where tables 7.2 - 7.4 reveals result obtained for single channel signals and tables 7.5 - 7.7 for multi-channel signals. Obtained results vary for various signals with different signal manipulations and de-synchronization attacks.

Experiment # 1: Performed on single channel sounds with around 20 signals and was found successful on all the signals.

Table 7.2: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 1	0	0.0200	0.0100	0.0200
Host Audio 2	0	0.0100	0.0000	0.0000
Host Audio 3	0	0.0110	0.0100	0.0100
Host Audio 4	0	0.0100	0.0000	0.0010

Table 7.3: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 1	0.0520	0.0110	0.0320	0.0210
Host Audio 2	0.0150	0.0110	0.0410	0.0615
Host Audio 3	0.0480	0.0100	0.0120	0.0413
Host Audio 4	0.0400	0.0100	0.0220	0.0125

Table 7.4: Robustness test for desynchronization attacks (in BER \times 100%)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 1	0.0150	0.0390	0.0150	0.0420
Host Audio 2	0.0130	0.0330	0.0600	0.0700
Host Audio 3	0.0200	0.0410	0.0400	0.0900
Host Audio 4	0.0462	0.0420	0.0290	0.0450

Experiment # 2: Performed on multi-channel sounds with around 33 signals and was found successful on all the signals.

Table 7.5: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 5	0	0.0130	0.0011	0.0240
Host Audio 6	0	0.0120	0.0010	0.0110
Host Audio 7	0	0.0220	0.0011	0.0410
Host Audio 8	0	0.0211	0.0040	0.0031

Table 7.6: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 5	0.0055	0.0220	0.0200	0.0170
Host Audio 6	0.0630	0.0230	0.0180	0.0370
Host Audio 7	0.0430	0.0014	0.0400	0.0300
Host Audio 8	0.0052	0.0021	0.0620	0.0210

Table 7.7: Robustness test for desynchronization attacks (in BER \times 100%)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 5	0.0400	0.0410	0.0360	0.0370
Host Audio 6	0.0390	0.0390	0.0470	0.0360
Host Audio 7	0.0410	0.0410	0.0290	0.0410
Host Audio 8	0.0310	0.0410	0.0380	0.0410

Evaluation of BER and recovery rate $(1 - BER) \times 100$ values helps in identifying the improved robustness of this watermarking algorithm compared to the prior one that does not employ any synchronization code in the watermark embedding.

Average of the recovery rate obtained for some signals are plotted below - figure 7.13:

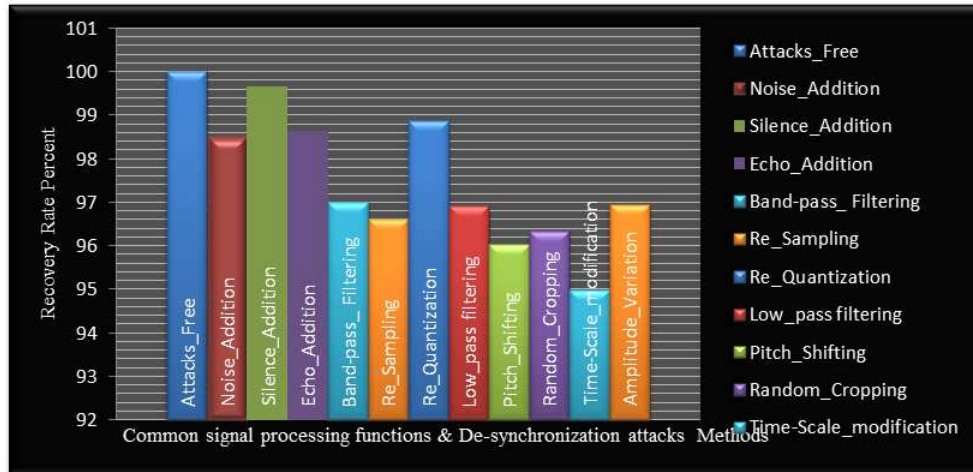


Figure 7.13: Average recovery rate

Capacity Tests

In this scheme, capacity test is conducted to determine the amount of watermark bits that are embedded and thus recovered in the audio stream in a unit time. Its evaluation follows the same equation 6.3.

In this work, different results were obtained due to the employment of different speech signals and its own dynamic watermark. For example, an audio signal with 3 seconds duration and 1278 bits watermark length

results in a capacity of 426 bps.

7.5 Summary

In this scheme, a variation of the prior watermarking method is suggested which utilize FFT and Barker code. The Barker code as synchronization code helps to improve the robustness nature of this scheme. Experimental results reveal that proposed scheme guarantees authenticity of the speech signal and also indicate that no compromise is made on audibility. It also reveals improvement in the proposed system in terms of robustness against various signal processing attacks such as noise addition, silence addition, echo addition, filtering and desynchronization attacks such as compression, cropping, amplitude variation and time-scale modification. Thus the newly proposed scheme eliminates some of the major downsides of the prior scheme. Use of synchronization code reduces the time of searching for the presence of FeatureMark bits and also eliminates the need to traverse through the entire acoustic signal for extracting the FeatureMark bits which in turn improves the performance of the system.

Chapter 8

FeatureMarking with QR Code

8.1 Introduction

This chapter demonstrates a different watermarking scheme employed towards the achievement of non-repudiation services using Fast Walsh- Hadamard Transform and Walsh code as the synchronization code. The proposed scheme is different from the prior schemes in terms of signal properties utilized as well as the prepared watermark. This scheme employs a different embedding and detection method.

8.2 Quick Response (QR) Code

As presented in the previous two chapters, encoding data as its 1-D and 2-D form can be achieved with barcodes and data matrix codes respectively. A data carrier represents data in a machine readable form; used to

enable automatic reading of the element strings. In this scheme QR Code is employed as watermark [Nerdery, Edwards, and Beecher 2010; Tumblr 2010; Encyclopedia 2013; DataMatrixCode 2010; Pontius 2012; QRworld 2011; Kaywa 2013b]. Nowadays, QR codes are extensively used in various industrial applications. A QR code is represented as a 2-D code placed on a white background and consists of black modules arranged in a square pattern. Any kind of data including binary or alphanumeric can be encoded to an online QR code generator. An online barcode scanner application either on a phone or PC can decode or display the details hidden in it.

Data matrix and QR code are both 2D codes that were developed around the same time. Data matrix was developed in the year 1989 and QR code in the year 1994. A data matrix barcode can hold up to 2,335 alphanumeric characters whereas QR code holds up to 4,296 alphanumeric characters. Data matrix code is believed to be the more secure (less hackable) code and is favored where high security is deemed important. Both data codes store considerably more information than the older 1D barcodes. For some time it seemed like data matrix was becoming the standard 2D barcode in North America as many organizations and various levels of government were using it. However data matrix wasn't designed to use Kanji (Japanese characters) and QR code was, so of course QR code became the prominent barcode throughout Japan.

8.3 Walsh Analysis

Fast Fourier transform or FFT is a common method for evaluating the transform domain spectra of a digital acoustic signal. In order to work with this, the time domain signals must be segmented into finite length blocks of sampled data called frames. Then, evaluate the length of the

frames in samples which is evaluated in units of time using the sampling rate.

Walsh transform, a discrete analog of the Fourier transform is employed in this scheme towards the mark embedding process [Oxford 2007; Schwengler 2013; Miranda 2002; CHOI 2000; Somogyi 2000]. Due to the fact that Walsh functions and its transforms are naturally more suited for digital computation, an effort is made to gradually replace the Fourier transform by Walsh type transforms. In this work, the digital watermark is embedded into the original voice signal by transforming it using FWHT. During the transmission process, FeatureMarked voice signal may suffer various signal manipulations including noise addition, silence addition, echo addition, re-sampling, re-quantization, low-pass and band-pass filtering or other desynchronization attacks such as amplitude variation, pitch shifting, random cropping and time-scale modification. Walsh transform is adopted for embedding the watermark under the assumption that for the case of random functions Walsh power spectra are slowly convergent and many Walsh transform components contain approximately equal signal power. Replacing such coefficients might not degrade the signal quality.

Amplitude Vs Frequency plot is represented in figure 8.1

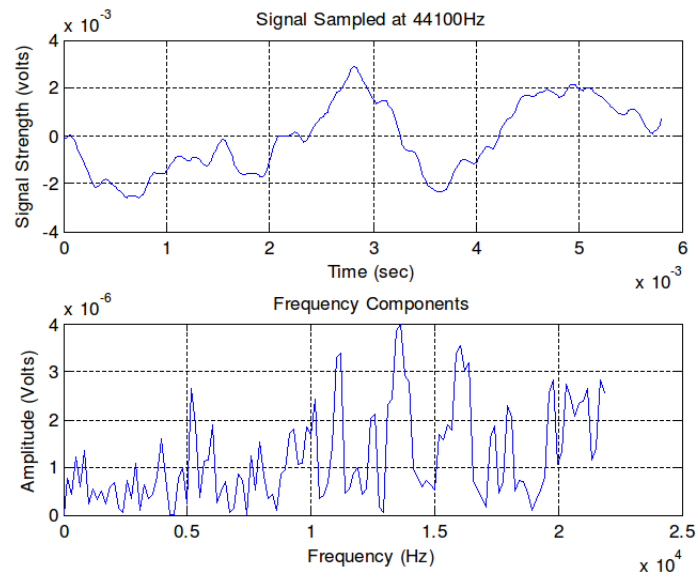


Figure 8.1: Amplitude-frequency plot

Walsh spectrum obtained by applying the FWHT function on the time-domain signal is represented as follows - figure 8.2:

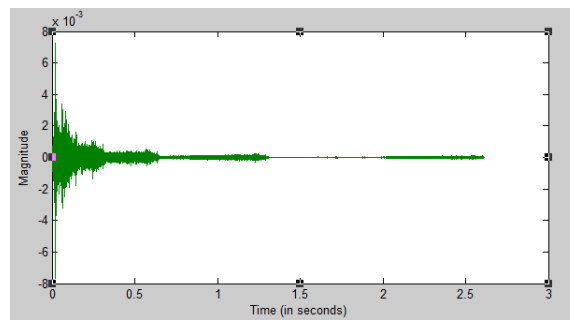


Figure 8.2: Walsh spectrum

After embedding watermark into the selected Walsh coefficients, its inverse function is employed to cancel the changes. As stated earlier, during the transmission process, voice signal may suffer some signal manipulations or desynchronization attacks. Embedding coefficients are selected in such a way that it could hold the robustness nature of the signal. Walsh transforms yields an order of the basis vectors with respect to their spectral properties. Robustness feature of the watermark can be achieved by selecting the Walsh coefficients in such a way that detection and reconstruction of the embedded watermark does not end in any degradation of the data [Tzafestas 1983; Zulfikar, Abbasi, and Alamoud 2013; Goresky and Klapper 2012].

Inverse Walsh transform in transform domain data recovers the original time domain signal. IFWHT is the function employed towards this process. ‘Symmetric flag’ helps to nullify the numerical inaccuracies, in other words to zero out the small imaginary components and to recover the original signal. Original time domain signal and the recovered time domain signal behave almost identically and does not reveal any significant difference while playing the audio.

Compared to the previous FFT schemes, it is possible to work with more FWHT coefficients and helps to embed more FeatureMark bits into it. Thus capacity of the system can be improved to a great extent. For example, Fourier transform of a signal employed in this system generates a matrix of size 139367×2 and its Walsh transform generates a matrix of size 262144×2 .

8.4 Proposed Scheme

In this scheme, original speech signal is divided into frames with a specific length l which is detailed in Chapter 4. FWHT is applied on these frames to

generate its magnitude spectrum. Magnitude values obtained are arranged in a matrix which is then transformed into its transform domain in order to embed the watermark bits.

Let $V = v(i), 0 \leq i < Length$ represent a host digital audio signal with Length samples. $FM = FM(i, j), 0 \leq i < M, 0 \leq j < N$ is a binary image to be embedded within the host audio signal, and $FM(i, j) \in \{0, 1\}$ is the pixel value at (i, j) .

8.4.1 Watermark Preparation

FeatureMark preparation module uses signal dependent physical features such as MFCC, spectral roll-off, spectral flux, spectral centroid and zero-cross rate that are obtained from the feature extraction module. Data code generated in this module is QR code and is embedded as the watermark.

Feature values obtained are quantized and arranged in a specific order to be submitted as input to an online QR code generator. Obtained QR code (figure 8.3) encodes all the quantized feature vectors of the recorded voice signal. Watermark in the proposed scheme is also stated as the FeatureMark.



Figure 8.3: Sample QR code

An enhanced scheme with an encryption method is also suggested to-

wards the preparation of the FeatureMark which is mentioned below:

$$y = x' \uplus_k \quad (8.1)$$

y is obtained by inserting $x' \bmod 7$ at the k^{th} position of x' , where $x' = x_1 x_2 \dots x_n x_{n+1} \dots x_{n+k}$ and $0 \leq k \leq m$ and $m = n + k$

This guarantees the security of the values that constitute the data code to a good extent.

8.4.2 Synchronization Code Generation

Synchronization code embedding is a crucial step associated with all audio watermarking schemes. Detection of watermark bits gets easy by embedding the synchronization code bits in the signal. Any variation in synchronization results in false detection and any spatial or transform domain modifications cause the detector lose its synchronization. Therefore it is appropriate to use right synchronization algorithms and robust synchronization code. In the proposed approach, a 16-bit Walsh code is embedded in front of respective FeatureMark to locate its position [Tanyel 2007; Miranda 2002; Encyclopedia 2013].

Walsh code is defined as a set of N codes, denoted W_j , for $j = 0, 1 \dots N-1$, which have the following properties:

- W_j takes on the values -1 or +1
- $W_j[0] = 1$ for all j
- W_j has exactly j zero crossings, for $j = 0, 1 \dots N-1$
- Each code W_j is either even or odd with respect to its mid-point
- The Walsh code used in this scheme is a 4×4 matrix

Employing the Matlab toolboxes for the generation of 4×4 Walsh code gives the following matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (8.2)$$

In order to embed these bits into the audio signal, the matrix is read row wise and then calculated its auto-correlation function. The result obtained is a 16-bit sequence and is employed as the synchronization code for the scheme that is introduced through this chapter.

8.4.3 Embedding Method

Embedding module in this scheme functions through three different levels: synchronization code embedding, FeatureMark embedding and repeat embedding.

First step associated with the embedding module is the segmentation where the original speech signal is divided into a set of segments and then each segment to two sub-segments. Secondly, with spatial watermarking technique, the Walsh code bits are embedded into the sub-segments. After embedding the synchronization code, subsequent segments are transformed using the fast Walsh transform.

Let $V = v(i), 0 \leq i < Length$ represent a host digital audio signal with Length samples. $FM = fm(i, j), 0 \leq i < M, 0 \leq j < N$ is a binary image to be embedded within the host audio signal, and $FM(i, j) \in 0, 1$ is the pixel value at (i, j) , $S = s(i), 0 \leq i \leq L_{syn}$ is a synchronization code with L_{syn} bits, where $s(i) \in 0, 1$.

Synchronization Code Embedding

Robust and transparent nature of audio watermarking schemes are achieved by embedding synchronization code bits in it. In this scheme, Walsh code is embedded in time domain samples as described below:

Algorithm 7 Synchronization code embedding

Steps

- 1: Each audio segment is divided into sub segments with n samples
 - 2: Consider the first audio segment $A_{10}(k)$, ($k = 0, 1, 2, \dots, L$) and its subsegments
 - 3: Take four consecutive samples in this segment, let it be s_1, s_2, s_3 and s_4 then
 - 4: **if** $s_1 + s_2 > s_3 + s_4$ **then**
 - 5: **if** $s_1 > s_2$ **then**
 - 6: insert 0 into s_1
 - 7: **else**
 - 8: insert 0 into s_2
 - 9: **end if**
 - 10: **else if** $s_3 + s_4 > s_1 + s_2$ **then**
 - 11: **if** $s_3 > s_4$ **then**
 - 12: insert 1 into s_3
 - 13: **else**
 - 14: insert 1 into s_4
 - 15: **end if**
 - 16: **else if** $s_1 + s_2 = s_3 + s_4$ **then**
 - 17: replace with 1 or 0
 - 18: **end if**
-

FeatureMark Embedding

Following figures (figure 8.4 & figure 8.5) illustrate the segmentation of an audio signal and the construction of embedding watermark bits respectively.

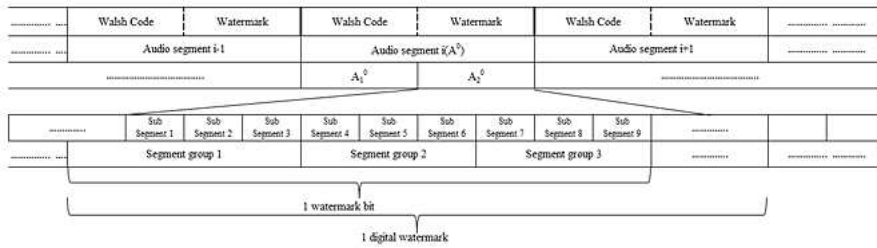


Figure 8.4: Audio segment and subsegment

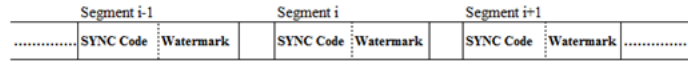


Figure 8.5: Construction of embedding information

In the FeatureMark embedding scheme, Arnold transform is applied to the generated FeatureMark image in order to dissipate its pixel space relationship. It helps in achieving improved strength of the extracted FeatureMark image. Scrambled image thus obtained is converted into a 1-dimensional sequence of 1s and 0s (binary digits) in order to embed the bits accurately into the 1-D audio signal.

Let $FM = FM(i, j), 0 \leq i \leq M, 0 \leq j \leq N$ represents the original FeatureMark image. Applying Arnold transform results in a scrambled structure which can be represented as figure 8.6.

$$FM_1 = FM_1(i, j), 0 \leq i \leq M, 0 \leq j \leq N.$$

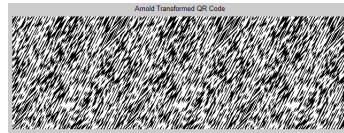


Figure 8.6: Arnold transformed QR code

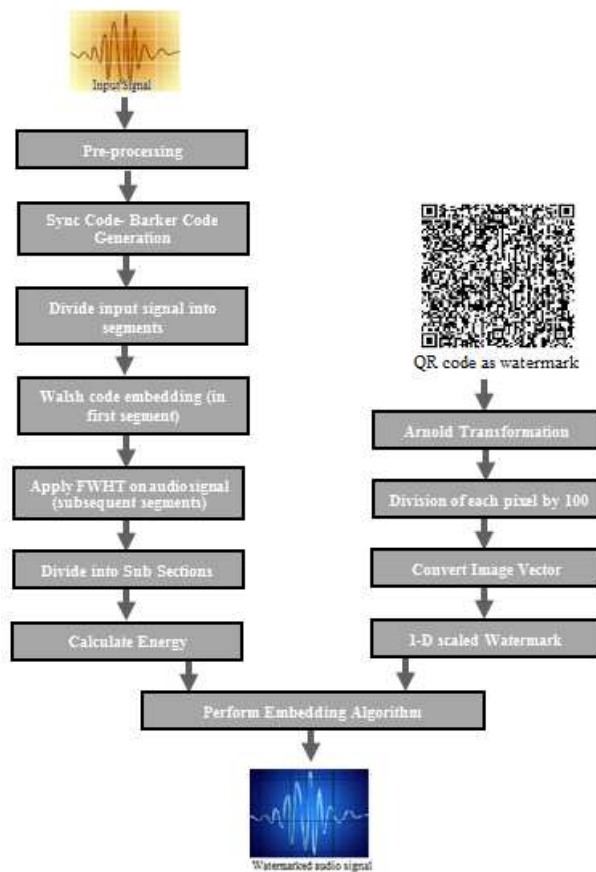


Figure 8.7: QR code embedding scheme

Then scrambled structure is converted into a sequence of 1s and 0s as follows:

$$FM_2 = fm_2(k) = FM_1(i, j), 0 \leq i \leq M, 0 \leq j \leq N,$$

$$k = i \times N + j, fm_2(k) \in \{0, 1\}$$

FeatureMark bits are then mapped into each signal frames by performing FWHT on the signal. Embedding scheme (figures 8.4 and 8.5) employs the following condition towards mapping of each FeatureMark bits. Overall system architecture on the embedding window can be summarized as - figure 8.7:

Algorithm 8 FeatureMark embedding

Inputs

- 1: Original speech signal
- 2: FeatureMark

Output

- 1: FeatureMarked speech signal

Steps

- 1: For the next subsegment consecutive to the SYNC embedded segment
 - Calculate the magnitude and phase spectrum of each subsegment using the FWHT : Let $f_1, f_2 \dots f_n$ be the magnitude coefficients obtained
 - Sort the magnitude coefficients in ascending order of their values: Let it be $f'_1, f'_2 \dots f'_n$
 - Energy entropy of each frame is then calculated using the following equation 8.3

$$I_j = - \sum_{i=1 \dots k} \sigma_i^2 \log_2 \sigma_i^2 \quad (8.3)$$

Algorithm 8 Featuremark embedding (continued)

- 2: Take four consecutive samples in this segment, let it be f_1, f_2, f_3, f_4 then
 - 3: **if** $f_1 + f_2 > f_3 + f_4$ **then**
 - 4: **if** $f_1 > f_2$ **then**
 - 5: insert 0 into f_1
 - 6: **else**
 - 7: insert 0 into f_2
 - 8: **end if**
 - 9: **else if** $f_3 + f_4 > f_1 + f_2$ **then**
 - 10: **if** $f_3 > f_4$ **then**
 - 11: insert 1 into f_3
 - 12: **else**
 - 13: insert 1 into f_4
 - 14: **end if**
 - 15: **else if** $f_1 + f_2 = f_3 + f_4$ **then**
 - 16: replace with 1 or 0
 - 17: **end if**
 - 18: Insert the watermark bits into these coefficients by using the following condition:
 - For embedding a 1 to f'_n , make $f''_n = f'_n + \alpha$
 - For embedding a 0 to f'_n , make $f''_n = f'_n - \alpha$, where $\alpha = 0.0001$
 - Repeat embedding
 - 19: Perform the same sequence on the subsequent segments till all the watermark bits are embedded
 - 20: Inverse FFT is applied to convert the frequency domain back to the original time domain to form the FeatureMarked speech signal
-

where, m is the length of the watermark sequence, f_i is a magnitude coefficient into which a watermark is embedded, x_i is a watermark to be inserted into f_i , α is a scaling factor and f'_i is an adjusted magnitude coefficient.

8.4.4 Repeat Embedding

Embedding the synchronization code bits and a single FeatureMark into an acoustic signal might not guarantee 100% robustness against common signal manipulations and desynchronization attacks. So these bits are repeatedly embedded into the selected coefficients which helps to reduce the number of ‘bit errors’ arouse due to these attacks. The precision or accuracy obtained for the extracted FeatureMark bits will be more with the use of ‘repeat embedding’.

8.4.5 Signal Reconstruction

Signal reconstruction is the final step in the embedding stage where transformed signal is converted back to time domain signal. For this, an inverse FWHT is applied. That is, modified spectra of the time domain signal is firstly converted using inverse FWHT. After this, combine each of the non-overlapping or overlapping window-frame series to reconstruct the FeatureMarked time domain signal.

From this it is understood that as in FFT transform, FWHT transform on the overlap regions is also not effective for embedding the FeatureMark bits since it may be thrown away during the signal reconstruction procedure. For example, consider two consecutive overlapping frames ‘m’ and ‘m+1’. Applying the function `fwht()` on ‘m’ generates the spectra of ‘m’ and any modification on these coefficients affects the subsequent m+1 frame’s coefficients. That is, the spectral modification applied to a particular frame will get distorted by the spectral modifications applied to its subsequent overlapping frame. This will result in reduced accuracy of the extracted FeatureMark bits as the modification on each frame represents the embedded FeatureMark bits.

For an FWHT based FeatureMarking scheme, magnitude spectrum of each frame is computed using the Matlab function `fwht()` on the original speech signal. Obtained values are arranged into a matrix of order $M \times N$ which is employed for embedding the FeatureMark bits.

8.4.6 FeatureMark Detection Scheme

Watermarking scheme employed in this method is blind, as it does not need the original speech signal or any other additional information for the detection of embedded watermark bits. The detection scheme functions as a two-step process where synchronization code detection is followed by the FeatureMark detection. FeatureMark bits are extracted by considering the fact that the bits present in between two synchronization code constitute the embedded FeatureMark.

Synchronization Code Detection

As in the prior scheme, synchronization code detection involves segmentation of the FeatureMarked signal. Then, for each sub-segment, the presence of Walsh code is confirmed. As the synchronization code used is Walsh code, the bits are embedded in the time domain and we can directly identify the existence of these bits in the subsegments.

The following figure demonstrates this procedure - figure 8.8:

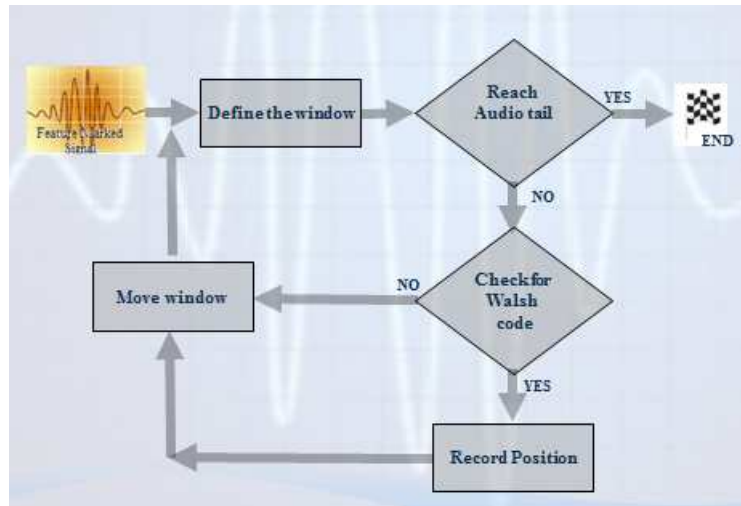


Figure 8.8: Walsh code detection

SYNC detection is implemented using the Algorithm 9.

FeatureMark Detection

Synchronization code detection is followed by the FeatureMark detection. FeatureMark bits are identified by evaluating the spectrum of each frame by FWHT. It is also seen that, FeatureMark bits are also followed by another set of synchronization code bits. Then mark bits are fetched out of the signal and are arranged in a matrix of order $M' \times N'$ to finalize the FeatureMark.

Algorithm 9 Synchronization code detection

Steps

- 1: First the FeatureMarked speech signal is divided into segments and subsegments
 - 2: Consider the first audio segment $A_{10}(k)$, ($k = 0, 1, 2, \dots, L$) and its subsegments
 - 3: Check for the presence of Walsh code bits in the time domain samples
 - 4: Take four consecutive samples in this segment, let it be s_1, s_2, s_3 and s_4 then
 - 5: **if** $s_1 + s_2 > s_3 + s_4$ **then**
 - 6: **if** $s_1 > s_2$ **then**
 - 7: extract 0 from s_1
 - 8: **else**
 - 9: extract 0 from s_2
 - 10: **end if**
 - 11: **else if** $s_3 + s_4 > s_1 + s_2$ **then**
 - 12: **if** $s_3 > s_4$ **then**
 - 13: extract 1 from s_3
 - 14: **else**
 - 15: extract 1 from s_4
 - 16: **end if**
 - 17: **else if** $s_1 + s_2 = s_3 + s_4$ **then**
 - 18: extract the bit 0 or 1
 - 19: **end if**
 - 20: Once the bits are extracted, a comparison should be done to avoid the false positiveness
-

8.4.7 Digital Watermark Extraction

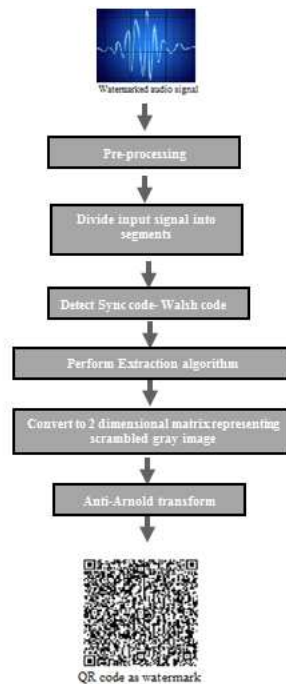


Figure 8.9: Watermark extraction scheme

The above algorithm demonstrates the extraction process (shown in figure 8.9) implemented for this scheme.

After confirming the presence of mark in the signal, FeatureMark bits are extracted by employing the conditions stated in the embedding module. Use of synchronization code helped to extract the FeatureMark bits directly out of the marked signal. According to this condition, FeatureMark bits are extracted out of the signal and the number of bits extracted should

Algorithm 10 FeatureMark extraction

Inputs

- 1: FeatureMarked speech signal

Output

- 1: FeatureMark

Steps

- 1: Consecutive subsegments of the FeatureMarked speech signal are transformed into FWHT domain.
 - 2: Do (for each subsegment)
 - 3: Take four consecutive samples in this segment, let it be f_1, f_2, f_3, f_4 then
 - 4: **if** $f_1 + f_2 > f_3 + f_4$ **then**
 - 5: **if** $f_1 > f_2$ **then**
 - 6: extract 0 from f_1
 - 7: **else**
 - 8: extract 0 from f_2
 - 9: **end if**
 - 10: **else if** $f_3 + f_4 > f_1 + f_2$ **then**
 - 11: **if** $f_3 > f_4$ **then**
 - 12: extract 1 from f_3
 - 13: **else**
 - 14: extract 1 from f_4
 - 15: **end if**
 - 16: **else if** $f_1 + f_2 = f_3 + f_4$ **then**
 - 17: extract the bit 0 or 1
 - 18: **end if**
 - 19: Convert the one-dimensional sequence into a two-dimensional scrambled structure
 - 20: Apply the anti-Arnold transform on this scrambled structure to obtain the original FeatureMark image
 - 21: Inverse FWHT is applied to convert the frequency domain back to the original time domain (if needed)
 - 22: Repeat the process if it is necessary to extract one more FeatureMark from the signal
-

match with that of the embedded FeatureMark bits.

8.5 An Enhanced FeatureMarking Method

An enhanced method is suggested which employs an encryption technique towards the preparation of its FeatureMark and is demonstrated below:

Algorithm 11 Enhanced FeatureMarking Method

Inputs

- 1: Original speech signal
- 2: FeatureMark

Output

- 1: FeatureMarked speech signal

Steps

- 1: Perform the pre-processing - framing and windowing
 - 2: Extract feature vectors using FFT and save it into a database - MFCCs, spectral flux, spectral roll-off, spectral centroid
-

Encryption scheme suggested guarantees a double layered security for the FeatureMark preparation. Watermark detection scheme employs the same technique to decrypt the original feature values. Though encryption scheme is limited to the final scheme can be incorporated in the watermark preparation module for all schemes proposed through this research, it is introduced as advancement over other schemes in order to guarantee a double layered security.

Algorithm 11 Enhanced FeatureMarking Method (continued)

3: Feature values extracted are given to an encryption scheme

$$y = x' \uplus_k \quad (8.4)$$

where, y is obtained by inserting $x' \bmod 7$ at the k^{th} position of x' , where $x' = x_1 x_2 \dots x_n x_{n+1} \dots x_{n+k}$ and $0 \leq k \leq m$ and $m = n + k$

- 4: Results obtained are given as input to an online QR code generator and the generated QR code will be termed as the FeatureMark
 - 5: Synchronization code embedding - This system involves insertion of orthogonal codes in the signal so as to appear at the beginning of each watermark. (Employs Algorithm 7)
 - 6: Embed the FeatureMark to the original signal by performing FWHT (Employs Algorithm 8)
 - 7: Perform the IFWHT and send the FeatureMarked signal to the desired recipient
 - 8: At the receiving end:
 - Detect the presence of the FeatureMark to confirm the authenticity of the received signal
 - Obtain the feature vectors from the extracted FeatureMark using an online QR Code scanner
 - Perform the decryption scheme to obtain the exact feature values
 - Feature comparison to confirm the authenticity of the voice signal
-

8.6 Experimental Results

Various experiments were conducted to evaluate the performance of this scheme by conducting imperceptibility tests, robustness tests and capacity tests. Around 25 Malayalam speech signals were collected from 10 members for this purpose. Signal characteristics such as 16 bits/sample with sampling rate of 44100 kHz are considered. Matlab version R2009b and music editor sound recorder is used for inducing common signal manipulations

and desynchronization attacks in the sample. Signal duration vary for each signal and is in the range of 2 - 300s. Number of frames employed towards signal processing depends on the length of the signal and with a frame rate of 100. From the experiments conducted, it is obvious that the embedding scheme works fine with most of the samples.

Prepared FeatureMarks: QR codes (figures 8.10 & 8.11)



Figure 8.10: Sample 1 - QR code



Figure 8.11: Sample 2 - encrypted QR code

Encryption Scheme: Extracted feature values are given to the encryption module that alters the values using the equation 8.4 which helps to hide the original data encoded in the data codes that are labelled in the signal.

For example, let

$\{-20.67280.29474 - 0.064930.34742 - 0.291390.20970 - 0.087330.02549 -$

0.14598 – 0.06361 – 0.06986 – 0.177920.005780.02388 – 0.04532 – 0.08209 –
0.09210 – 0.02231 – 0.039570.01991}

be the MFCC values extracted from a speech signal

Then execution of the encryption scheme results in the following sequence:

{–20.067280.529474–0.6064930.534742–0.1291390.520970–0.5087330.402549–
0.314598 – 0.506361 – 0.606986 – 0.4177920.3005780.302388 – 0.304532 –
0.108209 – 0.609210 – 0.002231 – 0.3039570.201991}

where, the first number after the decimal point is selected for inserting the modulus value. Obtained sequence is given as input to an online data code generator for generating the corresponding FeatureMark and in this case the QR Code.

Single channel Audio Signals

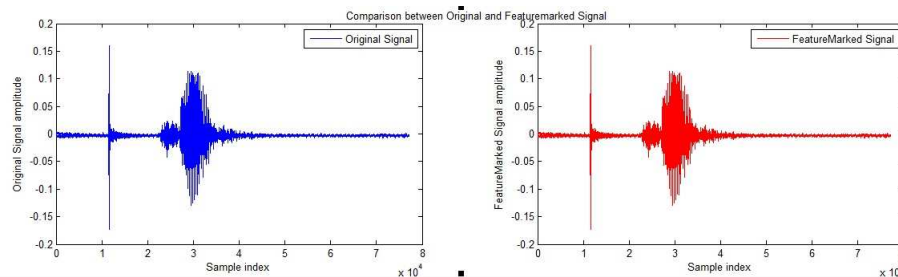


Figure 8.12: Single channel - original and FeatureMarked speech signal

With single channel audio signals (figure 8.12), selected FWHT coefficients are directly used for embedding the FeatureMark bits.

Multi Channel Audio Signals

In the case of stereo signals (figure 8.13) rather than embedding mark bits into two separate channels, the channels are added up as a single channel without affecting its signal properties.

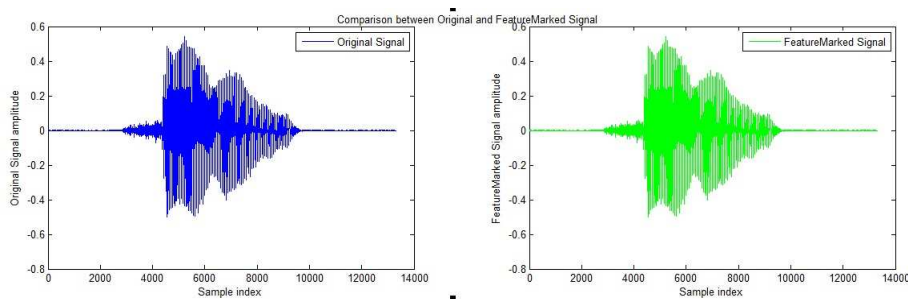


Figure 8.13: Multi-channel - original and FeatureMarked speech signal

Transparency Tests

As in prior methods, transparency of the proposed scheme is evaluated by employing subjective listening tests. Selection of listening panel is based on a set of non-expert listeners involving fellow research scholars and colleagues by considering the fact that non-expert listeners may be representative of the general population. This scheme demonstrates an excellent imperceptibility characteristic and the decision is based on the 5-point grade scale presented in table 6.1.

Imperceptibility criteria obtained for this scheme is presented in the below table 8.1:

Table 8.1: Imperceptibility criteria

Algorithm	Imperceptibility
FWHT & QR Code	Excellent

Robustness Tests

Robustness tests evaluate the FeatureMark detection accuracy against common signal manipulations and desynchronization attacks are conducted and are detailed below:

- Measure of strength of watermarking scheme against common signal processing functions is described in table 6.3

Common signal manipulations such as silence and echo addition, low-pass and band-pass filtering are performed using the music editor sound recorder. Others are done in Matlab. Robustness tests on common signal processing functions give a low value for BER.

- Measure of strength of watermarking scheme against desynchronization attacks is shown in the table 6.4.

Desynchronization attacks such as amplification to double volume and half volume, pitch change, speed changes and random cropping are also done using the music editor sound recorder.

Performance of the proposed FeatureMarking scheme is confirmed by assessing the BER values. The evaluation is performed on both single channel and multi-channel audio signals. Results obtained are given in the following tables.

Experiment # 1: Performed on single channel sounds with around 26 signals and was found successful on most of the signals.

Table 8.2: Robustness test for signal manipulations (in $\text{BER} \times 100\%$)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 1	0	0.0001	0.0001	0.0002
Host Audio 2	0	0.0000	0.0000	0.0001
Host Audio 3	0	0.0001	0.0003	0.0001
Host Audio 4	0	0.0002	0.0000	0.0001

Table 8.3: Robustness test for signal manipulations (in $\text{BER} \times 100\%$)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 1	0.0042	0.0001	0.0042	0.0039
Host Audio 2	0.0041	0.0001	0.0044	0.0035
Host Audio 3	0.0032	0.0002	0.0042	0.0043
Host Audio 4	0.0040	0.0002	0.0040	0.0040

Table 8.4: Robustness test for desynchronization attacks (in $\text{BER} \times 100\%$)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 1	0.0045	0.0039	0.0500	0.0042
Host Audio 2	0.0039	0.0033	0.0060	0.0037
Host Audio 3	0.0042	0.0041	0.0040	0.0039
Host Audio 4	0.0062	0.0042	0.0039	0.0045

Experiment # 2: Performed on multi-channel sounds with around 24 signals and was found successful on most of the signals.

Table 8.5: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Attacks Free	Noise Addition	Silence Addition	Echo Addition
Host Audio 5	0	0.0002	0.0001	0.0002
Host Audio 6	0	0.0000	0.0000	0.0001
Host Audio 7	0	0.0003	0.0001	0.0002
Host Audio 8	0	0.0001	0.0002	0.0003

Table 8.6: Robustness test for signal manipulations (in BER \times 100%)

Original Signal	Re-Sampling	Re-Quantization	Low-pass filtering	Band-pass Filtering
Host Audio 5	0.0040	0.0001	0.0040	0.0039
Host Audio 6	0.0038	0.0002	0.0039	0.0038
Host Audio 7	0.0055	0.0002	0.0050	0.0031
Host Audio 8	0.0033	0.0001	0.0700	0.0039

Table 8.7: Robustness test for desynchronization attacks (in BER \times 100%)

Original Signal	Amplitude Variation	Pitch Shifting	Random Cropping	Time-Scale modification
Host Audio 5	0.0020	0.0004	0.0400	0.0040
Host Audio 6	0.0040	0.0034	0.0050	0.0050
Host Audio 7	0.0004	0.0040	0.0020	0.0010
Host Audio 8	0.0037	0.0040	0.0039	0.0040

Average of the recovery rate obtained for some signals are plotted below - figure 8.14:

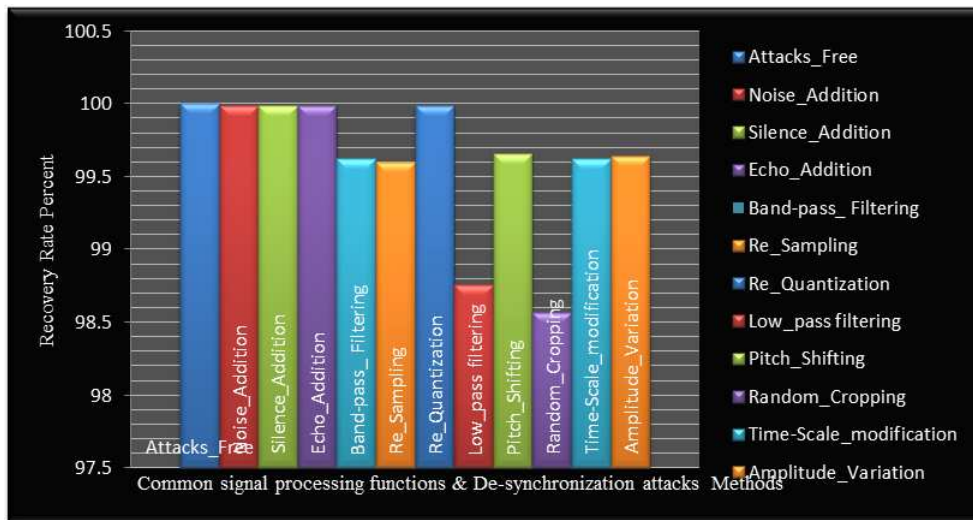


Figure 8.14: Average recovery rate

In this way, watermark bit error rate and recovery rate are evaluated to confirm the robust nature of the employed watermarking scheme. Recovery rate is calculated using the equation $(1 - BER) \times 100\%$ that denotes the knack of the proposed scheme in detection and reconstruction of the embedded watermark without any failure. The BER values obtained and recovery rate shows that the proposed scheme can be applied with the real-time applications involving audio/speech watermarking.

Capacity Tests

Like the previous two schemes, capacity is evaluated that determines the amount of information that can be embedded and recovered in the audio

stream. It can be evaluated using the equation 6.3

Capacity obtained for this new algorithm varies depending upon the length of the watermark and with the duration of the signal. For a sample audio signal with 3 seconds duration and 1560 bits watermark length, the results show a capacity of 520 bps.

8.6.1 The Goal: Guaranteeing Non-repudiation services

Experimental results obtained for the FFT and FWHT schemes assure that the proposed systems are strictly adhered to non-repudiation and authenticity criteria. Comparison of the original as well as the extracted FeatureMark at the recipient side helps to confirm authenticity of the signal. Feature extraction module can be carried out at the receiving end in case of any disputes to create the same FeatureMark and to perform a cross comparison. These in turn helps to achieve the non-repudiation services. Thus, the proposed scheme guarantees an authentic communication scheme for transferring audio or speech signals without causing any ownership disputes. Important characteristics of a non-repudiation scheme such as creating and storing the evidence of the sender and the receiver, fairness, timeliness and confidentiality is accomplished without any difficulty.

8.6.2 Merits and Demerits of the Proposed Schemes

The initial scheme suggested in chapter 6 guarantee the non-repudiation services to a good extend along with good embedding capacity. A draw back to this scheme is the time taken to detect the watermark bits. As an improvement to this a second scheme was introduced that employs the synchronization code embedding that helps to detect the mark bits without traversing the entire signal. Imperceptibility and robustness of this

scheme is improved but the capacity of this scheme is low. The last scheme presented introduces Walsh transform and the QR code that improves the imperceptibility, robustness and capacity of the embedding scheme. Another scheme that employs the encryption scheme in the mark preparation module provides a double-layered security.

8.7 Summary

In this scheme, a different watermarking method is suggested using FWHT for authenticating speech signals and to guarantee non-repudiation. The method employs Walsh code as its synchronization code and improves the robustness nature of the scheme. Experimental results reveal that the proposed scheme guarantees authenticity of the speech signal. It also indicates that the proposed scheme has not compromised audibility. The results also suggest robustness against various signal processing attacks such as noise addition, silence addition, echo addition, filtering and desynchronization attacks such as compression, cropping, amplitude variation and time-scale modification. The scheme overcomes downsides identified in the prior schemes proposed in Chapter 6, 7 and thus improves the time and computational complexity to a great extent.

Chapter 9

Conclusions and Future Works

9.1 Brief Summary

The dissertation lead us to one of the key focus area in audio watermarking and this chapter summarizes the work by describing a brief idea on what have been done for guaranteeing a secure audio communication and what comparative study has been performed with some of the existing watermarking schemes. It also includes contribution of the suggested work and some of the future enhancements that can be implemented to augment the system's security means. In order to develop a speaker authentication scheme that can guarantee non-repudiation, three varying watermarking schemes have been introduced that employ signal dependent dynamic watermarks. The thesis is able to report the smallest detail involving each steps in the proposed schemes. It takes us through the initial pre-processing step, feature extraction, speaker recognition, synchronization code genera-

tion, watermark preparation, synchronization code embedding, watermark embedding and finally the detection as well as the extraction methods.

Pre-processing involves framing and windowing techniques and help in improving the results of analysis. The key audio signal features such as MFCC, flux, roll-off, centroid, zero-cross rate, energy entropy, short-time energy and the fundamental frequency are evaluated as part of the feature extraction module. Extracted features are grouped for classification where the apt features are identified that help in the speaker recognition activity. The classifiers that have been employed include the ANN, k-NN and SVM. Features that have been decided from this module are utilized for the generation of signal dependent watermark using online data code generators. Watermarks used in the presented schemes include the barcode, data matrix code and QR code. Developed watermarks are embedded into the signal by transforming it using either FFT or FWHT. Embedding module chose different coefficients for different schemes. A total of three different schemes have been proposed which differs in the selection of embedding and thus the extracting coefficients as well as the data code employed towards it. First scheme employs FFT for its watermarking procedure. In an attempt to augment robustness and performance the subsequent scheme uses Barker Code as its synchronization code with the FFT signal transform . Final scheme apply contention solving technique utilizing Walsh code as the synchronization code and FWHT signal transform to cater all the weaknesses of previous schemes.

9.2 Comparison with Existing Schemes

A comparative study has been conducted to confirm the efficiency of the newly suggested watermarking schemes:

- Existing Schemes:
 - holds a generic or static watermark
 - most of the schemes in transform domain works with FFT
- Proposed Schemes:
 - holds signal dependent watermark
 - one among the few audio watermarking scheme that make use of Walsh transform
 - good imperceptibility
 - excellent robustness
 - good capacity

Table 9.1: Existing Watermarking Schemes

Algorithm	Imperceptibility	Robustness	Capacity
DFT for Stereo [2013]	Good	Satisfactory (3 %)	n/a
DWT & TSM [2011]	Good	Good	n/a
Wavelet Moment Invariance [2011]	Good	Strong	n/a
Pseudo-Zernike Moments [2011]	Good	Good	n/a
Pseudo Random Sequence [2005]	Good	Good	13 bps

Table 9.2: Proposed Watermarking Schemes

Algorithm	Imperceptibility	Robustness	Capacity
FFT & Barcode	Good	Good	17.22 bps
FFT & DM	Excellent	Good	11.48 bps
FWHT & QRCode	Excellent	Excellent	13.77 bps

9.3 Contributions

Study of audio watermarking has become significant due to its strategic as well as commercial importance. This dissertation addresses some of the key concerns in the current audio watermarking practices. Following are the salient highlights of this thesis:

- Speaker recognition with text-dependent and text-independent signals;
- Identification and verification of speakers are conducted using ANN, k-NN and SVM classifiers;
- Three different signal dependent watermarks termed as FeatureMark using barcode, datamatrix and QR code;
- Strength of the prepared FeatureMarks and the watermarking schemes are achieved by using a strong encryption scheme during the watermark generation;
- A speaker authentication scheme that employs FFT and barcode as the FeatureMark;
- A speaker authentication scheme that employs FFT and data matrix code as the FeatureMark;

-
- A varying authentication scheme that employs FWHT and QR code as the FeatureMark;
 - One among the few audio watermarking scheme that works with Walsh functions;
 - Speaker authentication methods that can be extended towards guaranteeing non-repudiation;
 - Implementation can be extended in real-time scenarios such as government, legal, banking and military services where audio authentication schemes are unavoidable.

9.4 Future Scope

Prospect augmentations on the proposed watermarking schemes are listed below:

- Human language impact on FeatureMarking scheme;
- Perform subjective listening test with a set of expert listeners;
- Consider this study for mimic sounds and its impact;
- Enhance the study for context dependency;
- Enhance the method to work with different file formats.

9.5 Summary

In this section an effort is taken to bring out prominent highlights of the work, the specific inferences on each of the proposed schemes and prospects

for future research in this area. The tests undoubtedly ascertain that the objectives of this work is satisfied and it can be assured that the proposed method functions as a secure, robust voice authentication system that guarantees a non-repudiated service for voice signal communication. The use of voice signal features, its classification and feature marking offers an improved scheme for authentic voice communication. Thus proposed audio watermarking schemes can outperform the existing techniques in terms of high success rates.

Appendix A

Notations and abbreviations used in the thesis

Notations:

- α : Alpha - Scaling factor
- Ω : Omega - True frequency in radians per second
- Π : Pi - The numerical value of Pi is 3.14159265358979323846...
- ζ : Zeta - Groups obtained by frame division
- φ : Varphi - Groups obtained by frame division
- ρ : Rho - Signal distortion
- \oplus : Bigoplus - A direct sum
- \uplus : Uplus - A mathematical operator used to represent equation 8.1

- \in : In - Element of
- Δ : Delta - Step size of the uniform scalar quantizer

Abbreviations:

- ACW : Adaptive Component Weighting
- ANN : Artificial Neural Networks
- ASP : Analog Signal Processing
- AWGN : Additive White Gaussian Noise
- BER : Bit Error Rate
- BPNN : Back Propagation Neural Network
- CDMA : Code Division Multiple Access
- CD : Compact Disc
- CPN : Counter Propagation Neural Network
- DCT : Discrete Cosine Transform
- DFRST : Discrete Fractional Sine Transform
- DFT : Discrete Fourier Transform
- DM : Dither Modulation
- DSP : Digital Signal Processing
- DSSS : Direct Sequence Spread Spectrum
- DTFT : Discrete-Time Fourier Transform
- DTW : Dynamic Time Warping

-
- DWT : Discrete Wavelet Transform
 - EE : Energy Entropy
 - EMD : Empirical Mode Decomposition
 - f_0 : Fundamental Frequency
 - FFT : FFT : Fast Fourier Transform
 - FM : FM : Feature Mark
 - FWHT : Fast Walsh Hadamard Transform
 - GMM : Gaussian Mixture Model
 - GPA : Generalized Patchwork Algorithm
 - GUI : Graphical User Interface
 - HAS : Human Auditory System
 - HMM : Hidden Markov Model
 - HVS : Human Visual System
 - IDTF : Inverse Discrete Fourier Transform
 - IDFT : Inverse Discrete Fourier Transform
 - IMF : Intrinsic Mode Function
 - ITU : International Telecommunication Union
 - JND : Just Noticeable Distortion
 - KFDA : Kernel Fisher Discriminant Analysis
 - k-NN : k-Nearest Neighbor
 - LCM : Log Coordinate Mapping
 - LPC : Linear Predictive Coding

- LWT : Lifting Wavelet Transform
- MAP : Maximum A-Posteriori
- MATLAB : Matrix Laboratory
- MCLT : Modulated Complex Lapped Transform
- MFCC : Mel-frequency Cepstral Coefficients
- ML : Maximum Likelihood
- MOS : Mean Opinion Score
- MPEG : Moving Picture Experts Group
- MPM : Multiplicative Patchwork Method
- MSE : Mean Squarred Error
- NMR : Noise to Mask Ratio
- ODG : Objective Difference Grade
- PDF : Probability Density Function
- PEAQ : Perceptual Evaluation of Audio Quality
- PFA : Prime-Factor (Good-Thomas) Algorithm
- PN : Pseudo Random Noise
- PRA : Pseudo Random Array
- QFT : Quick Fourier Transform
- QIM : Quantization Index Modulation
- QR : Quick Response Code
- RSVD : Reduced Singular Value Decomposition
- SC : Spectral Centroid

-
- SCD : Selective Correlation Detector
 - SDFCC : Speaker Dependent Frequency Cepstrum Coefficients
 - SE : Short-Time Energy
 - SF : Spectral Flux
 - SMR : Signal to Mask Ratio
 - SNR : Signal to Noise Ratio
 - SPL : Sound Pressure Level
 - SQAM : Sound Quality Assessment Material
 - SR : Spectral Roll-Off
 - SRE : Speaker Recognition Evaluation
 - SVD : Singular Value Decomposition
 - SVM : Support Vector Machine
 - TS : Time Spread
 - VQ : Vector Quantization
 - WAV : Waveform Audio File Format
 - WHC : Watermark-to-Host Correlation
 - ZCR : Zero Cross Rate

Appendix B

List of Publications

Part of the work presented in this thesis has been published/communicated to journals/digital libraries

1. Remya A R, A Sreekumar and Supriya M. H. “Comprehensive Non-repudiate Speech Communication Involving Geo-tagged FeatureMark”, Transactions on Engineering Technologies - World Congress on Engineering and Computer Science 2014, Springer Book. *Accepted*
2. Remya A R, A Sreekumar. “User Authentication Scheme Based on Fast-Walsh Hadamard Transform”, IEEE Digital Explore Library - 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT], Noorul Islam University (NIUEE), Thuckalay. 978-1-4799-7074-2/15/ ©2015 IEEE. *Accepted*
3. Remya A R, A Sreekumar. “An FWHT Based FeatureMarking Scheme for Non-repudiate Speech Communication”, Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress

- on Engineering and Computer Science 2014, 22-24 October, 2014, San Francisco, USA. ISBN: 978-988-19252-0-6. *Accepted*
4. Remya, A. R., et al. "An Improved Non-Repudiate Scheme-Feature Marking Voice Signal Communication." *International Journal of Computer Network & Information Security* 6.2 (2014)
 5. Remya, A. R., M. H. Supriya, and A. Sreekumar. "A Novel Non-repudiate Scheme with Voice FeatureMarking." *Computational Intelligence, Cyber Security and Computational Models*. Springer India, 2014. 183-194.
 6. Remya A R, A Sreekumar, "Voice Signal Authentication with Data matrix Code as the Watermark", *International Journal of Computer Networks and Security*, Recent Science, ISSN: 2051-6878, Vol.23, Issue.2, 2013
 7. Remya A R, A Sreekumar, "Authenticating Voice Communication with Barcode as the Watermark", *International Journal of Computer Science and Information Technologies*, Vol. 4 (4) , 2013, 560 - 563
 8. Remya A R, A Sreekumar, "An Inductive Approach to the Knack of Steganology", *International Journal of Computer Applications* (0975 - 8887), Volume 72 - No.15, June 2013
 9. Remya A R, A Sreekumar, "A Review on Indian Language Steganography", *CSI Digital Resource Center*, National Conference on Indian Language Computing NCILC'13

Part of the work include in the thesis has been presented in various national/international conferences

1. Remya A R, A Sreekumar. “User Authentication Scheme Based on Fast-Walsh Hadamard Transform“, IEEE Digital Explore Library - 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT], Noorul Islam University (NIUEE), Thuckalay. 978-1-4799-7074-2/15/ ©2015 IEEE *Accepted*
2. Remya A R, A Sreekumar. “An FWHT Based FeatureMarking Scheme for Non-repudiate Speech Communication”, International Conference on Computer Science and Applications - World Congress on Engineering and Computer Science (ICCSA'14 - WCECS) 2014, IAENG (ISBN: 978-988-19252-0-6), San Fransisco, US. *Accepted*
3. Remya, A. R., M. H. Supriya, and A. Sreekumar. “A Novel Non-repudiate Scheme with Voice FeatureMarking.” Computational Intelligence, Cyber Security and Computational Models. Springer India, 2014. 183-194.
4. Remya A R, A Sreekumar, “A Review on Indian Language Steganography”, CSI Digital Resource Center, National Conference on Indian Language Computing NCILC'13.

Bibliography

- [AA10] A. Ali and M. Ahmad. “Digital audio watermarking based on the discrete wavelets transform and singular value decomposition”. In: *European Journal of Scientific Research* 39.1 (2010), pp. 6–21.
- [Aer05] Aeroflex. *Introduction to FFT Analysis*. <http://www.aeroflex.com/>. 2005.
- [AHMB11] A. Al-Haj, A. Mohammad, and L. Bata. “DWT-Based Audio Watermarking.” In: *International Arab Journal of Information Technology (IAJIT)* 8.3 (2011).
- [AKM10] M. Akhaee, N. Kalantari, and F. Marvasti. “Robust audio and speech watermarking using Gaussian and Laplacian modeling”. In: *Signal processing* 90.8 (2010), pp. 2487–2497.
- [Alt08] R. e. a. Altalli. *Investigating the Existence of Barker Codes*. <http://www.math.wpi.edu/MPI2008/TSC/TSCeindlijk.pdf/>. 2008.

-
- [AM94] K. Assaleh and R. Mammone. “New LP-derived features for speaker identification”. In: *Speech and Audio Processing, IEEE Transactions on* 2.4 (1994), pp. 630–638.
- [AN+11] W. Al-Nuaimy et al. “An SVD audio watermarking approach using chaotic encrypted images”. In: *Digital Signal Processing* 21.6 (2011), pp. 764–779.
- [Ang11] X. Anguera. *Short-time analysis of speech*. http://www.xavieranguera.com/tdp_2011/5b-short-time_analysis.pdf/. 2011.
- [AP98] R. Anderson and F. Petitcolas. “On the limits of steganography”. In: *Selected Areas in Communications, IEEE Journal on* 16.4 (1998), pp. 474–481.
- [API76] H. Andrews and C. Patterson III. “Singular value decomposition (SVD) image coding”. In: *Communications, IEEE Transactions on* 24.4 (1976), pp. 425–432.
- [Arn00] M. Arnold. “Audio Watermarking: Features, Applications and Algorithms.” In: *IEEE International Conference on Multimedia and Expo (II)*. 2000, pp. 1013–1016.
- [Ars09] H. Arshad. *Embedded Implementation of Speech Recognition System*. <http://haroon.99k.org/windowing.html/>. 2009.
- [Arv03] J. Arvelius. *Barker Code*. <http://www.irf.se/~johan/matlab/MST-svn/barker.html/>. 2003.

- [AS02] M. Arnold and K. Schilz. “Quality evaluation of watermarked audio tracks”. In: *Electronic Imaging 2002*. International Society for Optics and Photonics. 2002, pp. 91–101.
- [Ave] M. Averkiou. *Digital Watermarking*.
- [Bag08] P. Baggenstoss. *MATLAB toolbox for HMM*. `class-specific.com/csf/html/doc/node91.html/`. 2008.
- [BB11] A. Bastani and E. Behbahani. “A Comparison Between Walsh-Hadamard and Fourier Analysis of the EEG Signals”. In: *International Journal of Engineering Science & Technology* 3.7 (2011).
- [BBP92] M. Barlow, I. Booth, and A. Parr. “The collection of two speaker recognition targeted speech databases”. In: *Proc. Australian Internat. Conf. Speech Science and Technology*. 1992, pp. 706–711.
- [BC07] J. Bullock and U. Conservatoire. “Libxtract: A lightweight library for audio feature extraction”. In: *Proceedings of the International Computer Music Conference*. Vol. 43. 2007.
- [Bel99] A. Bell. “The dynamic digital disk”. In: *Spectrum, IEEE* 36.10 (1999), pp. 28–35.
- [Ben+96] W. Bender et al. “Techniques for data hiding”. In: *IBM systems journal* 35.3.4 (1996), pp. 313–336.
- [BF08] J. Blackledge and O. Farooq. “Audio data verification and authentication using frequency modulation based watermarking”. In: (2008).

- [BHT63] B. Bogert, M. Healy, and J. Tukey. “The quefrency alanalysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking”. In: *Proceedings of the symposium on time series analysis*. Vol. 15. chapter. 1963, pp. 209–243.
- [Bio06] Biometrics.gov. *Speaker Recognition*. 14. <http://www.biometrics.gov/Documents/speakerrec.pdf/>. 2006.
- [BKSD10] V. Bhat K, I. Sengupta, and A. Das. “An adaptive audio watermarking based on the singular value decomposition in the wavelet domain”. In: *Digital Signal Processing* 20.6 (2010), pp. 1547–1558.
- [BL02] R. L. Brian Loker. *Approach to Real Time Encoding of Audio Samples – A DSP Realization of the MPEG Algorithm*. http://www.mp3-tech.org/programmer/docs/mp3encoder_dsp.pdf/. 2002.
- [Bla07] J. Blackledge. “Digital watermarking and self-authentication using chirp coding”. In: (2007).
- [BTH96] L. Boney, A. Tewfik, and K. Hamdy. “Digital watermarks for audio signals”. In: *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on*. IEEE. 1996, pp. 473–480.
- [Cam+06] W. Campbell et al. “SVM based speaker verification using a GMM supervector kernel and NAP variability compensation”. In: *Acoustics, Speech and Signal Processing, 2006. ICASSP*

- 2006 Proceedings. 2006 IEEE International Conference on.* Vol. 1. IEEE. 2006, pp. I–I.
- [CAM00] T. Cedric, R. Adi, and I. McLoughlin. “Data concealment in audio using a nonlinear frequency distribution of PRBS coded data and frequency-domain LSB insertion”. In: *TENCON 2000. Proceedings*. Vol. 1. IEEE. 2000, pp. 275–278.
- [Car+96] M. Carey et al. “Robust prosodic features for speaker identification”. In: *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. Vol. 3. IEEE. 1996, pp. 1800–1803.
- [CEaA12] M. Charfeddine, M. El’ arbi, and C. Amar. “A new DCT audio watermarking scheme based on preliminary MP3 study”. In: *Multimedia Tools and Applications* (2012), pp. 1–37.
- [Cha02] D. Chandra. “Digital image watermarking using singular value decomposition”. In: *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*. Vol. 3. IEEE. 2002, pp. III–264.
- [Che+13] S. Chen et al. “Adaptive audio watermarking via the optimization point of view on the wavelet-based entropy”. In: *Digital Signal Processing* 23.3 (2013), pp. 971–980.
- [Che93] B. Chen. *Speech Signal Representations*. http://berlin.csie.ntnu.edu.tw/Courses/SpeechRecognition/Lectures2013/SP2013F_Lecture10_SpeechSignalAnalysis.pdf/. 1993.

-
- [CHO00] B. CHOI. *Orthogonal Codes*. <http://www-mobile.ecs.soton.ac.uk/bjc97r/pnseq-1.1/node5.html/>. 2000.
- [Chu+07] K. Chung et al. “On SVD-based watermarking algorithm”. In: *Applied Mathematics and Computation* 188.1 (2007), pp. 54–57.
- [CLY96] C. Che, Q. Lin, and D. Yuk. “An HMM approach to text-prompted speaker verification”. In: *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. Vol. 2. IEEE. 1996, pp. 673–676.
- [Cox+02] I. Cox et al. *Digital watermarking*. Vol. 53. Springer, 2002.
- [Cox+07] I. Cox et al. *Digital watermarking and steganography*. Morgan Kaufmann, 2007.
- [Cox+97] I. Cox et al. “Secure spread spectrum watermarking for multimedia”. In: *Image Processing, IEEE Transactions on* 6.12 (1997), pp. 1673–1687.
- [CS05] N. Cvejic and T. Seppanen. “Increasing Robustness of LSB Audio Steganography by Reduced Distortion LSB Coding.” In: *J. UCS* 11.1 (2005), pp. 56–65.
- [CS96] T. Coffey and P. Saidha. “Non-repudiation with mandatory proof of receipt, [doi >10.1145/232335.232338]”. In: *ACM SIGCOMM Computer Communication Review* 26.1 (1996), pp. 6–17.

- [CSR06] W. Campbell, D. Sturim, and D. Reynolds. “Support vector machines using GMM supervectors for speaker verification”. In: *Signal Processing Letters, IEEE* 13.5 (2006), pp. 308–311.
- [CTL05] C. Chang, P. Tsai, and C. Lin. “SVD-based digital image watermarking scheme”. In: *Pattern Recognition Letters* 26.10 (2005), pp. 1577–1586.
- [Cve04] N. Cvejic. *Algorithms for audio watermarking and steganography*. Oulun yliopisto, 2004.
- [CW01] B. Chen and G. Wornell. “Quantization index modulation: A class of provably good methods for digital watermarking and information embedding”. In: *Information Theory, IEEE Transactions on* 47.4 (2001), pp. 1423–1443.
- [CWS10] C. Chang, H. Wang, and W. Shen. “Copyright-proving scheme for audio with counter-propagation neural networks”. In: *Digital Signal Processing* 20.4 (2010), pp. 1087–1101.
- [CX13] N. Chen and H. Xiao. “Perceptual audio hashing algorithm based on Zernike moment and maximum-likelihood watermark detection”. In: *Digital Signal Processing* 23.4 (2013), pp. 1216–1227.
- [Dat10] DataMatrixCode. *Datamatrix Code*. <http://www.datamatrixcode.net/>. 2010.
- [Dat12] T. P. Datacom. *Optimized Spectral Roll-Off*. http://www.paradisedata.com/collateral/articles/AN_0350optimised-SpectralRoll-OffApplicationNote.pdf/. 2012.

-
- [DDSN10] P. Diniz, E. Da Silva, and S. Netto. *Digital signal processing: system analysis and design*. Cambridge University Press, 2010.
- [DGP12] M. Dutta, P. Gupta, and V. Pathak. “A perceptible watermarking algorithm for audio signals”. In: *Multimedia Tools and Applications* (2012), pp. 1–23.
- [Dit+06] J. Dittmann et al. “Theoretical framework for a practical evaluation and comparison of audio watermarking schemes in the triangle of robustness, transparency and capacity”. In: *Transactions on Data Hiding and Multimedia Security I*. Springer, 2006, pp. 1–40.
- [DKJM10] P. Dhar, M. Khan, and K. Jong-Myon. “A new audio watermarking system using discrete fourier transform for copyright protection”. In: *International journal of computer science and network security* 6 (2010), pp. 35–40.
- [DLH95] D. DeFatta, J. Lucas, and W. Hodgkiss. *Digital signal processing : a system design approach*. Wiley. com, 1995.
- [DP09] A. Deshpande and K. Prabhu. “A substitution-by-interpolation algorithm for watermarking audio”. In: *Signal Processing* 89.2 (2009), pp. 218–225.
- [DWW03] S. Dumitrescu, X. Wu, and Z. Wang. “Detection of LSB steganography via sample pair analysis”. In: *Signal Processing, IEEE Transactions on* 51.7 (2003), pp. 1995–2007.
- [EB09] E. Ercelebi and L. Batakci. “Audio watermarking scheme based on embedding strategy in low frequency components with

- a binary image”. In: *Digital Signal Processing* 19.2 (2009), pp. 265–277.
- [Edw91] T. Edwards. “Discrete wavelet transforms: Theory and implementation”. In: *Universidad de* (1991).
- [EG02] J. Eggers and B. Girod. *Informed watermarking*. Springer, 2002.
- [Ell01] E. D. Ellis. “Design of a Speaker Recognition Code using MATLAB”. In: *Department of Computer and Electrical Engineering—University of Tennessee, Knoxville Tennessee 37996.09* (2001).
- [Enc13] Encyclopedia. *Speech and Speaker*. <http://en.wikipedia.org/wiki/>. 2013.
- [EP06] S. Emek and M. Pazarci. “Additive vs. image dependent DWT-DCT based watermarking”. In: *Multimedia content representation, classification and security*. Springer, 2006, pp. 98–105.
- [EPG07] Y. Erfani, M. Parviz, and S. Ghanbari. “Improved time spread echo hiding method for robust and transparent audio watermarking”. In: *Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th*. IEEE. 2007, pp. 1–4.
- [ES05] E. Ercelebi and A. Subasi. “Robust multi bit and high quality audio watermarking using pseudo-random sequences”. In: *Computers & Electrical Engineering* 31.8 (2005), pp. 525–536.
- [ES09] Y. Erfani and S. Siahpoush. “Robust audio watermarking using improved TS echo hiding”. In: *Digital Signal Processing* 19.5 (2009), pp. 809–814.

-
- [Fen04] L. Feng. “Speaker recognition”. PhD thesis. Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2004.
- [Fer+11] L. Ferrer et al. “Promoting robustness for speaker modeling in the community: the PRISM evaluation set”. In: *Proceedings of NIST 2011 Workshop*. 2011.
- [FGD00] J. Fridrich, M. Goljan, and R. Du. “Lossless data embeddingnew paradigm in digital watermarking”. In: *EURASIP Journal on Advances in Signal Processing* 2002.2 (1900), pp. 185–196.
- [FGD01] J. Fridrich, M Goljan, and R. Du. “Detecting LSB Steganography in Color and Gray-Scale Images”. In: *IEEE MultiMedia - doi >10.1109/93.959097* 8.4 (2001), pp. 22–28.
- [FIK06] R. Fujimoto, M. Iwaki, and T. Kiryu. “A method of high bit-rate data hiding in music using spline interpolation”. In: *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IHH-MSP’06. International Conference on*. IEEE. 2006, pp. 11–14.
- [Fil+07] J. Filipe et al. “Informatics in Control”. In: (2007).
- [FM09] M. Fallahpour and D. Megias. “High capacity audio watermarking using FFT amplitude interpolation”. In: *IEICE Electronics Express* 6.14 (2009), pp. 1057–1063.
- [FM10] M. Fallahpour and D. Megias. “DWT-based high capacity audio watermarking”. In: *IEICE transactions on fundamentals*

- of electronics, communications and computer sciences* 93.1 (2010), pp. 331–335.
- [FW09] M. Fan and H. Wang. “Chaos-based discrete fractional Sine transform domain audio watermarking scheme”. In: *Computers & Electrical Engineering* 35.3 (2009), pp. 506–516.
- [FW11] M. Fan and H. Wang. “Statistical characteristic-based robust audio watermarking for resolving playback speed modification”. In: *Digital Signal Processing* 21.1 (2011), pp. 110–117.
- [FWL08] M. Fan, H. Wang, and S. Li. “Restudy on SVD-based watermarking scheme”. In: *Applied Mathematics and Computation* 203.2 (2008), pp. 926–930.
- [FZHK06] M. Faundez-Zanuy, M. Haggmuller, and G. Kubin. “Speaker verification security improvement by means of speech watermarking”. In: *Speech communication* 48.12 (2006), pp. 1608–1619.
- [FZHK07] M. Faundez-Zanuy, M. Haggmuller, and G. Kubin. “Speaker identification security improvement by means of speech watermarking”. In: *Pattern Recognition* 40.11 (2007), pp. 3027–3034.
- [GFD01] M. Goljan, J. Fridrich, and R. Du. “Distortion-free data embedding for images”. In: *Information Hiding*. Springer. 2001, pp. 27–41.
- [Gho+04] J. Ghosh et al. “Automatic Speaker Recognition Using Neural Networks”. In: (2004).

-
- [Gia+06] T. Giannakopoulos et al. “Violence content classification using audio features”. In: *Advances in Artificial Intelligence*. Springer, 2006, pp. 502–507.
- [GK12] M. Goresky and A. Klapper. “Arithmetic Correlations and Walsh Transforms”. In: *IEEE Transactions On Information Theory* 58.1 (2012).
- [GLB96] D. Gruhl, A. Lu, and W. Bender. “Echo hiding”. In: *Information Hiding*. Springer. 1996, pp. 295–315.
- [GMS09] M. Gulbis, E. Muller, and M. Steinebach. “Content-based audio authentication watermarking”. In: *International journal of innovative computing, information and control* 5.7 (2009), pp. 1883–1892.
- [Gol01] Goldsmiths. *Musicians and Artists*. <http://doc.gold.ac.uk/~ma701rj/fe3.html/>. 2001.
- [Gop05a] K. Gopalan. “Audio steganography by cepstrum modification”. In: *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP'05). IEEE International Conference on*. Vol. 5. IEEE. 2005, pp. v–481.
- [Gop05b] K. Gopalan. “Robust watermarking of music signals by cepstrum modification”. In: *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE. 2005, pp. 4413–4416.

- [GREW11] D. Garcia-Romero and C. Espy-Wilson. “Analysis of i-vector Length Normalization in Speaker Recognition Systems.” In: *Interspeech*. 2011, pp. 249–252.
- [Gri08] D. Griffiths. *Head first statistics*. “O’Reilly Media, Inc.”, 2008.
- [GS110] GS1. *Introduction to GS 1 DataMatrix*. http://www.gs1.org/docs/barcodes/GS1_DataMatrix_Introduction_and_technical_overview.pdf/. 2010.
- [GS66] W. Gentleman and G. Sande. “Fast Fourier Transforms: for fun and profit”. In: *Proceedings of the November 7-10, 1966, fall joint computer conference*. ACM. 1966, pp. 563–578.
- [GW04] K. Gopalan and S. Wemndt. “Audio steganography for covert data transmission by imperceptible tone insertion”. In: *Proc. The IASTED International Conference on Communication Systems And Applications (CSA 2004), Banff, Canada*. 2004.
- [GZE03] E. Ganic, N. Zubair, and A. Eskicioglu. “An optimal watermarking scheme based on singular value decomposition”. In: *Proceedings of the IASTED International Conference on Communication, Network, and Information Security*. Vol. 85. 2003.
- [Ham08] A. R. Hambley. *Electrical Engineering: Principles & Applications*. Pearson Prentice Hall, 2008.
- [Har72] H. F. Harmuth. *Transmission of information by orthogonal functions*. Vol. 393. Springer-Verlag New York, 1972.

-
- [HC07] Y. Hu and Z. Chen. “An SVD-Based Watermarking Method for Image Authenticion”. In: *Machine Learning and Cybernetics, 2007 International Conference on*. Vol. 3. IEEE. 2007, pp. 1723–1728.
- [HC12] H. Hu and W. Chen. “A dual cepstrum-based watermarking scheme with self-synchronization”. In: *Signal Processing 92.4* (2012), pp. 1109–1116.
- [Hea10] R. Healy. “Digital audio watermarking for broadcast monitoring and content identification”. PhD thesis. National University of Ireland, Maynooth., 2010.
- [Hen+00] J. Hennebert et al. “POLYCOST: A telephone-speech database for speaker recognition”. In: *Speech communication* 31.2 (2000), pp. 265–270.
- [HKS06] A. Hatch, S. Kajarekar, and A. Stolcke. “Within-class covariance normalization for SVM-based speaker recognition.” In: *INTERSPEECH*. 2006.
- [HS06] X. He and M. Scordilis. “An enhanced psychoacoustic model based on the discrete wavelet packet transform”. In: *Journal of the Franklin Institute* 343.7 (2006), pp. 738–755.
- [Hus13] I. Hussain. “A novel approach of audio watermarking based on S-box transformation”. In: *Mathematical and Computer Modelling* 57.3 (2013), pp. 963–969.
- [Hyp06] Hyperphysics. *Fast Fourier Transforms*. <http://hyperphysics.phy-astr.gsu.edu/>. 2006.

-
- [IEE79] IEEE. *Programs for digital signal processing*. Inst. of Electr. and Electronics Engineers, 1979.
- [Inm+98] M. Inman et al. “Speaker identification using hidden Markov models”. In: *Signal Processing Proceedings, 1998. ICSP’98. 1998 Fourth International Conference on*. IEEE, 1998, pp. 609–612.
- [IP11a] V. Ingle and J. Proakis. *Digital signal processing using MATLAB*. Cengage Learning, 2011.
- [IP11b] V. Ingle and J. Proakis. *Digital Signal Processing Using MATLAB*. CengageBrain. com, 2011.
- [IR02] ITU-R. *Methodology for the subjective assessment of the quality of television pictures*. <http://www.itu.int/>. 2002.
- [IT96] ITU-T. *Methods for Subjective Determination of Transmission Quality*. International Telecommunication Union, 1996.
- [Jai89] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [Jay08] Jayshankar. “Efficient computation of the DFT of an $2N$ -point real sequence using FFT with CORDIC based butterflies”. In: *47. TENCON 2008 - 2008 IEEE Region 10 Conference*. IEEE, 2008, pp. 19–21.
- [Jin07] Q. Jin. “Robust speaker recognition”. PhD thesis. Carnegie Mellon University, 2007.

- [KAK07] N. Kalantari, S. Ahadi, and A. Kashi. “A robust audio watermarking scheme using mean quantization in the wavelet transform domain”. In: *Signal Processing and Information Technology, 2007 IEEE International Symposium on*. IEEE. 2007, pp. 198–201.
- [Kal+09] N. Kalantari et al. “Robust multiplicative patchwork method for audio watermarking”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 17.6 (2009), pp. 1133–1141.
- [Kav13] K. Kavitha. “An automatic speaker recognition system using MATLAB”. In: *World Journal of Science and Technology* 2.10 (2013).
- [Kay13a] Kaywa. *KaywaDataMatrix*. <http://datamatrix.kaywa.com/>. 2013.
- [Kay13b] Kaywa. *KaywaQRCode*. <http://qrcode.kaywa.com/>. 2013.
- [KB13] K. Khaldi and A. Boudraa. “Audio watermarking via EMD”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 21.3 (2013), pp. 675–680.
- [KD07] C. Kraetzer and J. Dittmann. “Mel-cepstrum-based steganalysis for VoIP steganography”. In: *Electronic Imaging 2007*. International Society for Optics and Photonics. 2007, pp. 650505–650505.
- [Kim+04] H. Kim et al. “Audio watermarking techniques”. In: *Intelligent Watermarking Techniques* 7 (2004), p. 185.

-
- [Kin05] T. H. Kinnunen. *Optimizing spectral feature based text-independent speaker recognition*. University of Joensuu, 2005.
- [KKS07] S. Krishna Kumar and T. Sreenivas. “Increased watermark-to-host correlation of uniform random phase watermarks in audio signals”. In: *Signal Processing* 87.1 (2007), pp. 61–67.
- [KL10] T. Kinnunen and H. Li. “An overview of text-independent speaker recognition: from features to supervectors”. In: *Speech communication* 52.1 (2010), pp. 12–40.
- [KM03] D. Kirovski and H. Malvar. “Spread-spectrum watermarking of audio signals”. In: *Signal Processing, IEEE Transactions on* 51.4 (2003), pp. 1020–1033.
- [KMB11] M. Keyvanpour and F. Merrikh-Bayat. “Robust dynamic block-based image watermarking in DWT domain”. In: *Procedia Computer Science* 3 (2011), pp. 238–242.
- [KNS02] B. Ko, R. Nishimura, and Y. Suzuki. “Time-spread echo method for digital audio watermarking using PN sequences”. In: *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. Vol. 2. IEEE. 2002, pp. II–2001.
- [KNS05a] H. Kameoka, T. Nishimoto, and S. Sagayama. “Harmonic-temporal structured clustering via deterministic annealing EM algorithm for audio feature extraction”. In: *Proc. ISMIR*. Vol. 2005. 2005.

-
- [KNS05b] B. Ko, R. Nishimura, and Y. Suzuki. “Time-spread echo method for digital audio watermarking”. In: *Multimedia, IEEE Transactions on* 7.2 (2005), pp. 212–221.
- [Kon+06] W. Kong et al. “SVD Based Blind Video Watermarking Algorithm.” In: *ICICIC (1)*. 2006, pp. 265–268.
- [Koy00] D Koya. “Aural phase distortion detection”. In: *Master’s thesis, University of Miami* (2000).
- [KP06] S. Katzenbeisser and F. Petitcolas. “Information Hiding Techniques for Steganography and Digital Watermarking”. In: (2006).
- [KP99] M. Kutter and F. Petitcolas. “Fair benchmark for image watermarking systems”. In: *Electronic Imaging ’99*. International Society for Optics and Photonics. 1999, pp. 226–239.
- [Kub95] G. Kubin. “What is a chaotic signal?” In: *IEEE Workshop on nonlinear signal and image processing*. Vol. 1. Citeseer. 1995, pp. 141–144.
- [KXZ10] M. Khan, L. Xie, and J. Zhang. “Chaos and NDFT-based spread spectrum concealing of fingerprint-biometric data into audio signals”. In: *Digital Signal Processing* 20.1 (2010), pp. 179–190.
- [KY01] K. Kaabneh and A. Youssef. “Muteness-based audio watermarking technique”. In: *Distributed Computing Systems Workshop, 2001 International Conference on*. IEEE. 2001, pp. 379–383.

-
- [KYH11] X. Kang, R. Yang, and J. Huang. “Geometric invariant audio watermarking based on an LCM feature”. In: *Multimedia, IEEE Transactions on* 13.2 (2011), pp. 181–190.
- [LC00] Y. Lee and L. Chen. “High capacity image steganographic model”. In: *IEE Proceedings-Vision, Image and Signal Processing* 147.3 (2000), pp. 288–294.
- [Lei+12] B. Lei et al. “A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition”. In: *Signal Processing* 92.9 (2012), pp. 1985–2001.
- [Lei02] W. J. Leis. *Digital Signal Processing: A MATLAB-based Tutorial Approach*. Research Studies Press Limited, 2002.
- [Lei11] J. W. Leis. *Digital signal processing using MATLAB for students and researchers*. John Wiley & Sons, 2011.
- [LHP96] L. Liu, J. He, and G. Palm. “Signal modeling for speaker identification”. In: *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. Vol. 2. IEEE. 1996, pp. 665–668.
- [Li+11] H. Li et al. “Audio Binary Halftone Watermarking Algorithm”. In: *Procedia Engineering* 15 (2011), pp. 2700–2704.
- [Lin01] P.-L. Lin. “Digital watermarking models for resolving rightful ownership and authenticating legitimate customer”. In: *Journal of Systems and Software* 55.3 (2001), pp. 261–271.

-
- [LJZ01] L. Lu, H. Jiang, and H. Zhang. “A robust audio classification and segmentation method”. In: *Proceedings of the ninth ACM international conference on Multimedia*. ACM. 2001, pp. 203–211.
- [LNK06] J. Liu, X. Niu, and W. Kong. “Image Watermarking Based on Singular Value Decomposition”. In: *III-MSP*. 2006, pp. 457–460.
- [LPV82] S. Lipshitz, M. Pocock, and J. Vanderkooy. “On the audibility of midrange phase distortion in audio systems”. In: *Journal of the Audio Engineering Society* 30.9 (1982), pp. 580–595.
- [LS04] Y. Liu and J. Smith. “Watermarking sinusoidal audio representations by quantization index modulation in multiple frequencies”. In: *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*. Vol. 5. IEEE. 2004, pp. V–373.
- [LS12] B. Lei and Y. Soon. “A multipurpose audio watermarking algorithm with synchronization and encryption”. In: *Journal of Zhejiang University SCIENCE C* 13.1 (2012), pp. 11–19.
- [LSL11] B. Lei, I. Soon, and Z. Li. “Blind and robust audio watermarking scheme based on SVD–DCT”. In: *Signal Processing* 91.8 (2011), pp. 1973–1984.
- [LSR13] B. Lei, I. Song, and S. Rahman. “Robust and secure watermarking scheme for breath sound”. In: *Journal of Systems and Software* 86.6 (2013), pp. 1638–1649.

- [LT02] R. Liu and T. Tan. “An SVD-based watermarking scheme for protecting rightful ownership”. In: *Multimedia, IEEE Transactions on* 4.1 (2002), pp. 121–128.
- [LT07] O. Lartillot and P. Toivainen. “A matlab toolbox for musical feature extraction from audio”. In: *International Conference on Digital Audio Effects*. 2007, pp. 237–244.
- [LWC98] Z. Liu, Y. Wang, and T. Chen. “Audio feature extraction and analysis for scene segmentation and classification”. In: *Journal of VLSI signal processing systems for signal, image and video technology* 20.1-2 (1998), pp. 61–79.
- [Lyo09] J. Lyons. *Practical Cryptography*. <http://practicalcryptography.com/>. 2009.
- [LZB04] Z. Linghua, Y. Zhen, and Z. Baoyu. “A new method to train VQ codebook for HMM-based speaker identification”. In: *Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*. Vol. 1. IEEE. 2004, pp. 651–654.
- [LZJ02] L. Lu, H. Zhang, and H. Jiang. “Content analysis for audio classification and segmentation”. In: *Speech and Audio Processing, IEEE Transactions on* 10.7 (2002), pp. 504–516.
- [MAK08] H. Malik, R. Ansari, and A. Khokhar. “Robust audio watermarking using frequency-selective spread spectrum”. In: *IET Information Security* 2.4 (2008), pp. 129–150.
- [Mal99] H. Malvar. “A modulated complex lapped transform and its applications to audio processing”. In: *Acoustics, Speech, and*

- Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. Vol. 3. IEEE. 1999, pp. 1421–1424.
- [Mar02] J. Martinez. “Standards-MPEG-7 overview of MPEG-7 description tools, part 2”. In: *MultiMedia, IEEE* 9.3 (2002), pp. 83–93.
- [MAS08] A. Mohammad, A. Alhaj, and S. Shaltaf. “An improved SVD-based watermarking scheme for protecting rightful ownership”. In: *Signal Processing* 88.9 (2008), pp. 2158–2180.
- [Mat+10] B. Mathieu et al. “YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software.” In: *ISMIR*. 2010, pp. 441–446.
- [MAT09] MAT. *Audio Feature Extraction*. <http://www.mat.ucsb.edu/201A/Notes/AudioFeatureextraction.html/>. 2009.
- [Mat84] Mathworks. *MATLAB*. <http://www.mathworks.com/>. 1984.
- [MCL08] V. Martin, M. Chabert, and B. Lacaze. “An interpolation-based watermarking scheme”. In: *Signal Processing* 88.3 (2008), pp. 539–557.
- [McL09] I. McLoughlin. *Applied speech and audio processing: with Matlab examples*. Cambridge University Press, 2009.
- [Mel99] H. Melin. “Databases for speaker recognition: Activities in COST250 working group 2”. In: *COST 250-Speaker Recognition in Telephony, Final Report 1999* (1999).

- [MF03] H. Malvar and D. Florencio. “Improved spread spectrum: a new modulation technique for robust watermarking”. In: *Signal Processing, IEEE Transactions on* 51.4 (2003), pp. 898–905.
- [MFD05] C. McKay, I. Fujinaga, and P. Depalle. “jAudio: A feature extraction library”. In: *Int. Conf. on Music Information Retrieval (ISMIR05)*. 2005, pp. 600–3.
- [MHJM03] D. Megias, J. Herrera-Joancomarti, and J. Minguillon. “A robust audio watermarking scheme based on MPEG 1 layer 3 compression”. In: *Communications and Multimedia Security. Advanced Techniques for Network and Data Protection*. Springer, 2003, pp. 226–238.
- [MHJM05] D. Megias, J. Herrera-Joancomarti, and J. Minguillon. “Total disclosure of the embedding and detection algorithms for a secure digital watermarking scheme for audio”. In: *Information and Communications Security*. Springer, 2005, pp. 427–440.
- [Mir02] H. Miranda. *Pseudo Noise Sequences*. http://paginas.fe.up.pt/~hmiranda/cm/Pseudo_Noise_Sequences.pdf/. 2002.
- [MM05] I. Mierswa and K. Morik. “Automatic feature extraction for classifying audio data”. In: *Machine learning* 58.2-3 (2005), pp. 127–149.
- [Mob98] B. G. Mobasser. “Direct sequence watermarking of digital video using m-frames”. In: *Image Processing, 1998. ICIP 98*.

- Proceedings. 1998 International Conference on*. Vol. 2. IEEE. 1998, pp. 399–403.
- [MSRF10] D. Megias, J. Serra-Ruiz, and M. Fallahpour. “Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification”. In: *Signal Processing* 90.12 (2010), pp. 3078–3092.
- [Mus11] MusicTech. *Audio Signal Levels Tutorial*. <http://www.musictech.net/2011/03/10mm-187-audio-signal-levels-explained/>. 2011.
- [Nat+12] I. Natgunanathan et al. “Robust patchwork-based embedding and decoding scheme for digital audio watermarking”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 20.8 (2012), pp. 2232–2239.
- [Nat+13] I. Natgunanathan et al. “Robust patchwork-based watermarking method for stereo audio signals”. In: *Multimedia Tools and Applications* (2013), pp. 1–24.
- [Nav14] D. R. Nave. *Loudness*. <http://hyperphysics.phy-astr.gsu.edu/hbase/sound/loud.html/>. 2014.
- [NEB10] T. Nerderly, M. Edwards, and F. Beecher. *Quick Response Codes*. <http://shouldiuseaqrcode.com/>. 2010.
- [Ned04] C. Nedeljko. “Algorithms for audio watermarking and steganography”. In: *Acta Universitatis Ouluensis. Series C* (2004).
- [Net+95] J. Neto et al. “Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system”. In: (1995).

- [Nis12] R. Nishimura. “Audio Watermarking Using Spatial Masking and Ambisonics”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 20.9 (2012), pp. 2461–2469.
- [Oh+01] H. Oh et al. “New echo embedding technique for robust and imperceptible audio watermarking”. In: *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP’01). 2001 IEEE International Conference on*. Vol. 3. IEEE. 2001, pp. 1341–1344.
- [Oni+04] J. Onieva et al. “Agent-mediated non-repudiation protocols”. In: *Electronic Commerce Research and Applications* 3.2 (2004), pp. 152–162.
- [Ori10] N. Orio. “Automatic identification of audio recordings based on statistical modeling”. In: *Signal Processing* 90.4 (2010), pp. 1064–1076.
- [Oro+08] I. Orovic et al. “Speech signals protection via logo watermarking based on the time–frequency analysis”. In: *annals of telecommunications-Annales des telecommunications* 63.7-8 (2008), pp. 369–377.
- [Ors10] F. Orsag. “Speaker Dependent Coefficients for Speaker Recognition”. In: *International Journal of Security & Its Applications* 4.1 (2010).
- [OSM05] H. Ozer, B. Sankur, and N. Memon. “An SVD-based audio watermarking technique”. In: *Proceedings of the 7th workshop on Multimedia and security*. ACM. 2005, pp. 51–56.

- [Oxf07] Oxford. *Orthogonal, Channelization, Scrambling and carrier modulation codes*. <http://www.dauniv.ac.in/downloads/Mobilecomputing/MobileCompChap-04L10Codingmthods.pdf/>. 2007.
- [OZL04] J. Onieva, J. Zhou, and J. Lopez. “Non-repudiation protocols for multiple entities”. In: *Computer Communications* 27.16 (2004), pp. 1608–1616.
- [PAK98] F. Petitcolas, R. Anderson, and M. Kuhn. “Attacks on copyright marking systems”. In: *Information Hiding*. Springer. 1998, pp. 218–238.
- [Pel+00] J. Pelecanos et al. “Vector quantization based Gaussian modeling for speaker verification”. In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. Vol. 3. IEEE. 2000, pp. 294–297.
- [Pen+13] H. Peng et al. “A learning-based audio watermarking scheme using kernel Fisher discriminant analysis”. In: *Digital Signal Processing* 23.1 (2013), pp. 382–389.
- [PG90] R. Peacocke and D. Graf. “An introduction to speech and speaker recognition”. In: *Computer* 23.8 (1990), pp. 26–33.
- [PK11] S. Palani and D. Kalaiyarasi. *Digital Signal Processing : for C.S.I.T.* Ane Books Pvt. Ltd, 2011.
- [Pla07] C. Plack. *Auditory Perception*. http://socialscientist.us/nphs/psychIB/psychpdfs/PIP_Auditory_Perception.pdf/. 2007.

- [Pon12] N. Pontius. *Datamatrix Codes Vs QR Codes*. <http://www.camcode.com/asset-tags/barcodes-data-matrix-vs-qr-codes/>. 2012.
- [PP08] B Prasad and S. Prasanna. *Speech, audio, image and biomedical signal processing using neural networks*. Springer, 2008.
- [PTW07] C. Park, D. Thapa, and G. Wang. “Speech authentication system using digital watermarking and pattern recovery”. In: *Pattern Recognition Letters* 28.8 (2007), pp. 931–938.
- [PW02] A. Paquet and R. Ward. “Wavelet-based digital watermarking for image authentication”. In: *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*. Vol. 2. IEEE. 2002, pp. 879–884.
- [PY02] V. Pathangay and B. Yegnanarayana. “Use of Vertical Face Profiles for Text Dependent Audio-Visual Biometric Person Authentication.” In: *ICVGIP*. 2002.
- [QRw11] QRworld. *QR Codes Versus Data Matrix*. <http://qrworld.wordpress.com/2011/09/26/qr-codes-versus-data-matrix/>. 2011.
- [Qua02] T. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. http://www.cs.tut.fi/kurssit/SGN-4010/ikkunointi_en.pdf/. 2002.
- [QZ04] X. Quan and H. Zhang. “Audio watermarking based on psychoacoustic model and adaptive wavelet packets”. In: *Signal*

- Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*. Vol. 3. IEEE. 2004, pp. 2518–2521.
- [Rab12] L. Rabiner. *Time Domain Methods in Speech Processing*. http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/digital-speechprocessingcourse/lectures_new/Lectures7-8_winter_2012.pdf/. 2012.
- [Rao+07] R. Rao et al. “Text-dependent speaker recognition system for Indian languages”. In: *International Journal of Computer Science and Network Security (IJCSNS 2007)* 7.11 (2007), pp. 65–71.
- [RE82] K. Rao and D. Elliott. “Fast transforms algorithms, analyses, applications”. In: *Acad. Press, New York, London* (1982).
- [Rei11] T. Reinhardt. *Barcode Generator*. <http://www.barcode-generator.org/>. 2011.
- [Rey02] D. A. Reynolds. “An overview of automatic speaker recognition”. In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Washington, DC: IEEE Computer Society. 2002, pp. 4072–4075.
- [RJ05] M. Russell and P. Jackson. “A multiple-level linear/linear segmental HMM with a formant-based intermediate layer”. In: *Computer Speech & Language* 19.2 (2005), pp. 205–225.
- [RMK07] A. Ranade, S. Mahabalarao, and S. Kale. “A variation on SVD based image compression”. In: *Image and Vision Computing* 25.6 (2007), pp. 771–777.

- [S12] S. S. *Audio Analysis - time domain /FFT /Spectrogram, National Instruments*. <https://decibel.ni.com/content/docs/DOC-20515/>. 2012.
- [Sca11] Scandit. *Barcode Types*. <http://www.scandit.com/2011/11/29/barcode-types-qr-codes-datamatrix-and-proprietary-2d-codes/>. 2011.
- [SCB05] A. Solomonoff, W. Campbell, and I. Boardman. “Advances in channel compensation for SVM speaker recognition”. In: *Proc. ICASSP*. Vol. 1. 2005, pp. 629–632.
- [Sch13] T. Schwengler. *CDMA*. <http://morse.colorado.edu/~tlen5510/text/classwebch8.html/>. 2013.
- [Sch98] S. Schneider. “Formal analysis of a non-repudiation protocol”. In: *Computer Security Foundations Workshop, 1998. Proceedings. 11th IEEE*. IEEE. 1998, pp. 54–65.
- [SD03] M. Steinebach and J. Dittmann. “Watermarking-based digital audio data authentication”. In: *EURASIP Journal on Applied Signal Processing* 2003 (2003), pp. 1001–1015.
- [SEL10] A. Science and A. Engineering Laboratories. *Speech, Non-Speech and Silence*. https://www.asel.udel.edu/speech/reports/ANN_class/ANN_class.html/. 2010.
- [SF12] A. Stolcke and M. Ferrer. “Effects of audio and ASR quality on cepstral and high-level speaker verification systems”. In: *Odyssey 2012-The Speaker and Language Recognition Workshop*. 2012.

-
- [SG86] A. Syrdal and H. Gopal. “A perceptual model of vowel recognition based on the auditory representation of American English vowels”. In: *The Journal of the Acoustical Society of America* 79.4 (1986), pp. 1086–1100.
- [SG90] M. Savic and S. Gupta. “Variable parameter speaker verification system based on hidden Markov modeling”. In: *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE. 1990, pp. 281–284.
- [SGD12] J. Singh, P. Garg, and A. De. “Audio watermarking based on quantization index modulation using combined perceptual masking”. In: *Multimedia tools and Applications* 59.3 (2012), pp. 921–939.
- [SH01] J. Seok and J. Hong. “Audio watermarking for copyright protection of digital audio data”. In: *Electronics Letters* 37.1 (2001), pp. 60–61.
- [SHK02] J. Seok, J. Hong, and J. Kim. “A novel audio watermarking algorithm for copyright protection of digital audio”. In: *etri Journal* 24.3 (2002), pp. 181–189.
- [SLD02] M. Steinebach, A. Lang, and J. Dittmann. “Stirmark benchmark: audio watermarking attacks based on lossy compression”. In: *Electronic Imaging 2002*. International Society for Optics and Photonics. 2002, pp. 79–90.

- [Som00] S. Somogyi. *Orthogonal Codes in CDMA: Generation and Simulation*. <http://www.ece.ualberta.ca/~elliott/ee552/studentApp-Notes/2000f/misc/CDMA/>. 2000.
- [SOV12] SOVARRwiki. *Spectral Roll-Off*. http://sovarr.c4dm.eecs.qmul.ac.uk/wiki/Spectral_Rolloff/. 2012.
- [SPA06] A. Spanias, T. Painter, and V. Atti. *Audio signal processing and coding*. John Wiley & Sons, 2006.
- [SQC04] A. Solomonoff, C. Quillen, and W. Campbell. “Channel compensation for SVM speaker recognition”. In: *Proceedings of Odyssey-04, The Speaker and Language Recognition Workshop*. 2004, pp. 57–62.
- [SS12] M. Sahidullah and G. Saha. “Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition”. In: *Speech Communication* 54.4 (2012), pp. 543–565.
- [SSY02] R. Sun, H. Sun, and T. Yao. “A SVD-and quantization based semi-fragile watermarking technique for image authentication”. In: *Signal Processing, 2002 6th International Conference on*. Vol. 2. IEEE. 2002, pp. 1592–1595.
- [Ste+01] M. Steinebach et al. “StirMark benchmark: audio watermarking attacks”. In: *Information Technology: Coding and Computing, 2001. Proceedings. International Conference on*. IEEE. 2001, pp. 49–54.

-
- [Stu11] G. L. Stuber. *Principles of mobile communication*. Springer, 2011.
- [Suz+95] Y. Suzuki et al. “An optimum computer-generated pulse signal suitable for the measurement of very long impulse responses”. In: *The Journal of the Acoustical Society of America* 97.2 (1995), pp. 1119–1123.
- [Swa+98] M. Swanson et al. “Robust audio watermarking using perceptual masking”. In: *Signal Processing* 66.3 (1998), pp. 337–355.
- [Tac+02] R. Tachibana et al. “An audio watermarking method using a two-dimensional pseudo-random array”. In: *Signal Processing* 82.10 (2002), pp. 1455–1469.
- [Tan07] M. Tanyel. “Comparing the Walsh domain to the Fourier domain with a labview based communication systems toolkit”. In: (2007).
- [TBI97] L. Trefethen and D. Bau III. *Numerical linear algebra*. Vol. 50. Siam, 1997.
- [TC05] H. Tsai and J. Cheng. “Adaptive signal-dependent audio watermarking based on human auditory system and neural networks”. In: *Applied Intelligence* 23.3 (2005), pp. 191–206.
- [TI12] TEC-IT. *Data Matrix (ECC200) - 2D Barcode*. <http://www.tec-it.com/en/support/knowledge/symbologies/datamatrix/Default.aspx/>. 2012.

- [TNS05] A. Takahashi, R. Nishimura, and Y. Suzuki. “Multiple watermarks for stereo audio signals using phase-modulation techniques”. In: *Signal Processing, IEEE Transactions on* 53.2 (2005), pp. 806–815.
- [Tum10] Tumblr. *WTF QR Codes*. <http://wtfqrcodes.com/>. 2010.
- [Tza04] G. Tzanetakis. *ICME 2004 Tutorial: Audio Feature Extraction*. <http://webhome.cs.uvic.ca/~gtzan/work/talks/icme04/icme04tutorial.pdf/>. 2004.
- [Tza83] S. Tzafestas. “Walsh transform theory and its application to systems analysis and control: an overview”. In: *Mathematics and Computers in Simulation* 25.3 (1983), pp. 214–225.
- [UKR07] K. Umapathy, S. Krishnan, and R. Rao. “Audio signal feature extraction and classification using local discriminant bases”. In: *Audio, Speech, and Language Processing, IEEE Transactions on* 15.4 (2007), pp. 1236–1246.
- [Uni04] A. Universitet. *Framing and deframing*. http://kom.aau.dk/group/04gr742/pdf/framing_worksheet.pdf/. 2004.
- [Vie+05] R. Vieru et al. “New results using the audio watermarking based on wavelet transform”. In: *Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on*. Vol. 2. IEEE. 2005, pp. 441–444.
- [Vis08] V. Viswanathan. “Information hiding in wave files through frequency domain”. In: *Applied Mathematics and Computation* 201.1 (2008), pp. 121–127.

-
- [VK+05] K. Von Kriegstein et al. “Interaction of face and voice areas during speaker recognition”. In: *Journal of cognitive neuroscience* 17.3 (2005), pp. 367–376.
- [VKG06] K. Von Kriegstein and A. Giraud. “Implicit multisensory associations influence voice recognition”. In: *PLoS biology* 4.10 (2006), e326.
- [VLG08] D. Varodayan, Y. Lin, and B. Girod. “Audio authentication based on distributed source coding”. In: *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE. 2008, pp. 225–228.
- [Wan] H. min Wang. *Speech Signal Representations, Part 2: Speech Signal Processing*. <http://www.iis.sinica.edu.tw/~whm/course/Speech-NTUT-2004S/slides/SpeechSignalRepresentations-p2.pdf/>.
- [Wan+08] H. Wang et al. “Fuzzy self-adaptive digital audio watermarking based on time-spread echo hiding”. In: *Applied Acoustics* 69.10 (2008), pp. 868–874.
- [Wan11] J. Wang. “New Digital Audio Watermarking Algorithms for Copyright Protection”. PhD thesis. National University of Ireland Maynooth, 2011.
- [WCC04] C. Wang, T. Chen, and W. Chao. “A new audio watermarking based on modified discrete cosine transform of MPEG/Audio Layer III”. In: *Networking, Sensing and Control, 2004 IEEE International Conference on*. Vol. 2. IEEE. 2004, pp. 984–989.

-
- [WF10] H. Wang and M. Fan. “Centroid-based semi-fragile audio watermarking in hybrid domain”. In: *Science China Information Sciences* 53.3 (2010), pp. 619–633.
- [WIP] WIPO. *The Impact of the Internet on Intellectual Property Law*. http://www.wipo.int/copyright/en/ecommerce/ip_survey/chap3.html/.
- [WK+10] H. Wook Kim et al. “Selective correlation detector for additive spread spectrum watermarking in transform domain”. In: *Signal Processing* 90.8 (2010), pp. 2605–2610.
- [WM97] D. Williams and V. Madisetti. *Digital signal processing Handbook*. CRC Press, Inc., 1997.
- [WMN11] X. Wang, T. Ma, and P. Niu. “A pseudo-Zernike moment based audio watermarking scheme robust against desynchronization attacks”. In: *Computers & Electrical Engineering* 37.4 (2011), pp. 425–443.
- [WNL11] X. Wang, P. Niu, and M. Lu. “A robust digital audio watermarking scheme using wavelet moment invariance”. In: *Journal of Systems and Software* 84.8 (2011), pp. 1408–1421.
- [WNQ08] X. Wang, P. Niu, and W. Qi. “A new adaptive digital audio watermarking based on support vector machine”. In: *Journal of Network and Computer Applications* 31.4 (2008), pp. 735–749.

-
- [WNY09] X. Wang, P. Niu, and H. Yang. “A robust digital audio watermarking based on statistics characteristics”. In: *Pattern Recognition* 42.11 (2009), pp. 3057–3064.
- [Wol11] Wolfram. *Signal Processing*. <http://reference.wolfram.com/mathematica/guide/SignalProcessing.html/>. 2011.
- [WPD99] R. Wolfgang, C. Podilchuk, and E. Delp. “Perceptual watermarks for digital images and video”. In: *Proceedings of the IEEE* 87.7 (1999), pp. 1108–1126.
- [WSK99] C. Wu, P. Su, and C. Kuo. “Robust audio watermarking for copyright protection”. In: *SPIE’s International Symposium on Optical Science, Engineering, and Instrumentation*. International Society for Optics and Photonics. 1999, pp. 387–397.
- [Wu+11] T. Wu et al. “Audio watermarking scheme with dynamic adjustment in mute period”. In: *Expert Systems with Applications* 38.6 (2011), pp. 6787–6792.
- [WZ06] X. Wang and H. Zhao. “A novel synchronization invariant audio watermarking scheme based on DWT and DCT”. In: *Signal Processing, IEEE Transactions on* 54.12 (2006), pp. 4835–4840.
- [XF02] C. Xu and D. Feng. “Robust and efficient content-based digital audio watermarking”. In: *Multimedia Systems* 8.5 (2002), pp. 353–368.
- [Xia+02] B. Xiang et al. “Short-time Gaussianization for robust speaker verification”. In: *Acoustics, Speech, and Signal Processing (ICASSP)*,

- 2002 *IEEE International Conference on*. Vol. 1. IEEE. 2002, pp. I-681.
- [Xia+11] Y. Xiang et al. “Effective pseudonoise sequence and decoding function for imperceptibility and robustness enhancement in time-spread echo-based audio watermarking”. In: *Multimedia, IEEE Transactions on* 13.1 (2011), pp. 2–13.
- [Xia+12] Y. Xiang et al. “A dual-channel time-spread echo method for audio watermarking”. In: *Information Forensics and Security, IEEE Transactions on* 7.2 (2012), pp. 383–392.
- [Xia11] S. Xiang. “Audio watermarking robust against D/A and A/D conversions”. In: *EURASIP Journal on Advances in Signal Processing* 2011.1 (2011), pp. 1–14.
- [Xio+03] Z. Xiong et al. “Comparing MFCC and MPEG-7 audio features for feature extraction, maximum likelihood HMM and entropic prior HMM for sports audio classification”. In: *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP’03). 2003 IEEE International Conference on*. Vol. 5. IEEE. 2003, pp. V-628.
- [Xio+06] Y. Xiong et al. “Covert communication audio watermarking algorithm based on LSB”. In: *Communication Technology, 2006. ICCT’06. International Conference on*. IEEE. 2006, pp. 1–4.

-
- [XKH08] S. Xiang, H. Kim, and J. Huang. “Audio watermarking robust against time-scale modification and MP3 compression”. In: *Signal Processing* 88.10 (2008), pp. 2372–2387.
- [Xu+99] C. Xu et al. “Applications of digital watermarking technology in audio signals”. In: *Journal of the Audio Engineering society* 47.10 (1999), pp. 805–812.
- [YCA97] Y. Yardimci, A. Cetin, and R. Ansari. “Data hiding in speech using phase coding”. In: *Fifth European Conference on Speech Communication and Technology*. 1997.
- [YK03a] I. Yeo and H. Kim. “Generalized patchwork algorithm for image watermarking”. In: *Multimedia systems* 9.3 (2003), pp. 261–265.
- [YK03b] I. Yeo and H. Kim. “Modified patchwork algorithm: A novel audio watermarking scheme”. In: *Speech and Audio Processing, IEEE Transactions on* 11.4 (2003), pp. 381–386.
- [YK99] C. Yeh and C. Kuo. “Digital watermarking through quasi m-arrays”. In: *Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on*. IEEE. 1999, pp. 456–461.
- [YMO95] K. Yu, J. Mason, and J. Oglesby. “Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation”. In: *IEE Proceedings-Vision, Image and Signal Processing* 142.5 (1995), pp. 313–318.
- [YO10] Z. Yucel and A. B. Ozguler. “Watermarking via zero assigned filter banks”. In: *Signal Processing* 90.2 (2010), pp. 467–479.

-
- [Yos+11] Y. Yoshitomi et al. “An Authentication Method for Digital Audio Using a Discrete Wavelet Transform”. In: *Journal of Information Security* 2 (2011), p. 59.
- [YWM11] H. Yang, X. Wang, and T. Ma. “A robust digital audio watermarking using higher-order statistics”. In: *AEU-International Journal of Electronics and Communications* 65.6 (2011), pp. 560–568.
- [YXG10] H. Yang, S. Xingming, and S. Guang. “A Semi-Fragile Watermarking Algorithm using Adaptive Least Significant Bit Substitution.” In: *Information Technology Journal* 9.1 (2010).
- [ZAA13] Z. Zulfikar, S. Abbasi, and A. Alamoud. “Design of Real Time Walsh Transform for Processing of Multiple Digital Signals”. In: *International Journal of Electrical and Computer Engineering (IJECE)* 3.2 (2013), pp. 197–206.
- [ZG96a] J. Zhou and D. Gollmann. “A fair non-repudiation protocol”. In: *IEEE symposium on security and privacy*. Citeseer. 1996, pp. 55–61.
- [ZG96b] J. Zhou and D. Gollmann. “Observations on non-repudiation”. In: *Advances in Cryptology - ASIACRYPT'96*. Springer. 1996, pp. 133–144.
- [ZG97a] J. Zhou and D. Gollmann. “An efficient non-repudiation protocol”. In: *Computer Security Foundations Workshop, 1997. Proceedings., 10th*. IEEE. 1997, pp. 126–132.

-
- [ZG97b] J. Zhou and D. Gollmann. “Evidence and non-repudiation”. In: *Journal of Network and Computer Applications* 20.3 (1997), pp. 267–281.
- [Zil01] R. D. Zilca. “Text-independent speaker verification using covariance modeling”. In: *Signal Processing Letters, IEEE* 8.4 (2001), pp. 97–99.
- [ZL05] X. Zhang and K. Li. “Comments on “An SVD-based watermarking scheme for protecting rightful Ownership””. In: *Multimedia, IEEE Transactions on* 7.3 (2005), pp. 593–594.
- [ZL99] J. Zhou and K. Lam. “Securing digital signatures for non-repudiation”. In: *Computer Communications* 22.8 (1999), pp. 710–716.

Index

- Imperceptibility tests, 4
- Acoustic properties, 20
- Acoustic signals, 100
- Acoustical properties, 19
- analog format, 18
- Analog processors, 18
- ANN, 113
- Anti-Arnold transform, 12
- Arnold transform, 12, 152, 177, 204
- Audio records, 19
- Audio signal, 34
- Audio signals, 17, 20
 - aperiodic, 19
 - features, 22
 - dynamic, 22
 - perceptual, 22, 29
 - physical, 22
 - static, 22
 - periodic, 19
 - properties, 19
 - spectral, 19
 - temporal, 19
 - representation, 20
 - auditory representation, 20, 21
 - spectrogram, 20, 21
- Audio waveforms, 19
- Auditory perception, 31
- Auditory scene analysis, 18
- Capacity tests, 4
- CDMA, 40
- Cognitive functions, 19
- Compression algorithm, 34
- Compression algorithms, 31
- Content based algorithm, 53
- Contributions, 228
- Cooley, Turkey algorithm, 99
- Critical band, 33
- Critical band analysis, 31
- Critical bands, 31
- Data-codes, 4
- digital format, 18
- Discrete sampling, 35

-
- DTW, 117
 - Fast-Fourier transformation, 99
 - Feature Extraction, 98
 - energy entropy, 107
 - frequency, 110
 - MFCC, 100
 - short-time energy, 108
 - spectral centroid, 106
 - spectral flux, 103
 - spectral roll-off, 104
 - ZCR, 110
 - Filtering, 90
 - Fourier Analysis, 145
 - Frame based analysis, 90
 - Frequency Component Analysis of Signals, 35
 - Continuous Fourier Transform, 36
 - Discrete Fourier Transform, 37
 - Discrete-Time Fourier Transform, 37
 - Fast Fourier Transform, 38
 - Fast Hadamard Transform, 41
 - Fourier Transform, 36
 - Hadamard Transform, 39
 - Frequency masking, 31, 32
 - Front-end processing, 122
 - Future Scope, 229
 - Global masking threshold, 33
 - Hadamard transform, 40
 - Harmonics, 89
 - HAS, 15, 18, 19, 32
 - NMR, 33
 - frequency masking, 32
 - SMR, 33
 - SNR, 33
 - temporal masking, 34
 - Hearing perception, 20
 - Hearing System, 18
 - HMM-GM, 114
 - HVS, 31
 - Imposter modeling, 123
 - kNN, 113
 - LPC, 117
 - MPEG-7, 21
 - Non-repudiation, 2
 - Non-repudiation Service, 169
 - Objectives, 3
 - One-dimensional & two-dimensional
 - Data Codes, 150
 - barcode, 150
 - Data matrix codes, 172

- QR Codes, 196
- Perceptual attributes, 19
- Post-masking, 34
- Power spectra, 31
- Pre-masking, 34
- Pre-Processing, 89
 - De-framing, 96
 - frame shifting, 96
 - frame shifting, 96
 - framing, 92
 - windowing, 92, 95
 - Hamming Window, 93
 - Hamming window, 93
 - Rectangular Window, 93
 - window function, 94
- Problem Statement, 3
- Proposed Non-repudiation Schemes
 - Scheme 1, 151
 - Scheme 1
 - Algorithm 1, 153
 - Algorithm 2, 158
 - Scheme 2, 172
 - Algorithm 1, 176
 - Algorithm 2, 178
 - Algorithm 3, 182
 - Algorithm 4, 185
 - Scheme 3, 200
 - Algorithm 1, 203
 - Algorithm 3, 209
 - Algorithm 4, 212
 - Algorithm 5, 214
- Railing function, 35
- Robustness tests, 4
- Sampling rate, 18
- Scope, 4
- SDFCC, 117
- Sharpness, 89
- Short-time analysis, 20
 - short-time parameters, 20
- Signal models, 19
- Signal power, 33
- Signal processing, 16
 - analog signal processing, 16
 - digital signal processing, 16
- Simultaneous masking, 31, 32
- Sound, 18, 19
- Speaker modeling, 122
- Speaker Recognition, 113, 114
 - ANN, 124
 - Comparative Study, 138
 - kNN, 130
 - identification approach, 135
 - recognition approach, 137
 - verification approach, 134
 - SVM, 131

- identification approach, 135
- recognition approach, 137
- verification approach, 134
- text-dependent, 121
- Speaker recognition evaluations, 118
- Speaker Verification, 122
- Speech and Audio Signal Processing, 34
 - sampling, 35
 - amplitude quantization, 35
 - impulses, 35
 - zero-order hold, 35
- Speech Signal, 89
- Speech signal, 34
- SRE, 118
- Steganography, 99
- SVM, 113
- Synchronization Code, 174
 - Barker code, 174
 - Walsh Code, 201
- Temporal masking, 31
- Testing, 48
- Transparency tests, 4
- Walsh Analysis, 197
- Watermarking, 41, 99
 - embedder, 45
 - audio watermarking, 41
 - communications model, 43
 - detector, 45
 - evaluation, 46
 - accuracy, 79
 - capacity, 80
 - computational efficiency, 80
 - imperceptibility, 78
 - robustness, 79
 - extraction, 45
 - general model, 42
 - geometric model, 44
 - statistical model, 43
- Watermarking Algorithms, 50
 - hybrid schemes, 65
 - chirp coding, 65
 - interpolation, 69
 - IPR, 74
 - muteness, 72
 - other schemes, 76
 - patchwork, 67
 - quantization, 65
 - SVD, 69
- time-domain based schemes, 51
 - LSB coding, 51
 - amplitude masking, 55
 - echo hiding, 53
 - phase coding, 55

transform-domain based schemes,

56

DCT based, 57

DWT based, 60

FFT based, 57

spread-spectrum based, 62

WG, 118

Working Group, 118

Zero-order hold, 34

ZOH, 34