A STUDY ON AN EFFICIENT TAMPERING DETECTION AND LOCALIZATION METHOD FOR SPEECH SIGNALS

By

Sharr Wint Yee Myint
B.C.Tech. (Hons.)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Technology
(M.C.Tech.)

University of Computer Studies, Yangon
November 2018

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude and appreciation to all persons who have encouraged, supported, and helped me in any aspect directly or indirectly during the completion of the thesis.

I would like to express my deepest gratitude to Dr. Mie Mie Thet Thwin, Rector of the University of Computer Studies, Yangon, for her kind permission to submit this thesis.

I am also grateful to Dr. Khin Than Mya, Professor and Head of the Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her helpful advice and administrative support throughout the development of the thesis.

I would also like to express my sincere thanks to Dr. Myat Thida Mon, Professor and Head of the Faculty of Computer Systems and Technologies, University of Information Technology, and Dr. Win Pa Pa, Associate Professor, Natural Language Processing Lab, University of Computer Studies, Yangon, and Daw Hnin Hnin Aye, Associate Professor, Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for agreeing to be on my dissertation committee. They are more than generous with their expertise and precious time for reviewing my thesis.

I am also very thankful to Dr. Thet Thet Khin, Associate Professor and Dean of the Master Course of the University of Computer Studies, Yangon, for her invaluable guidance and administrative support throughout the development of the thesis.

I would also like to express my deepest appreciation and respect to my supervisor, Dr. Twe Ta Oo, Assistant Lecturer, Faculty of Computer Systems and Technologies, for her helpful suggestion regarding the thesis topic and detailed guidance throughout the preparation of my thesis. Her broad knowledge is essential for me to enrich my ideas and constant encouragement has helped me to get a deep insight in the field of watermarking and then overcome many difficulties encountered during my thesis.

I would also like to give my appreciation and sincere honor to Daw Ni Ni San, Lecturer, Department of Language, University of Computer Studies, Yangon, for kindly editing my thesis from the language point of view.

I am also very grateful to all of my teachers from the University of Computer Studies, Yangon, for their insightful comments, valuable suggestions, helpful hints, fair criticisms, and fullest cooperation during the seminars of my thesis.

Last but not least, I would like to dedicate this thesis to my parents, who believe in me and have been thorough all the rough times. I also want to thank my beloved entire family and friends for their love and encouragements in both spiritually and materially at every stage of my personal and academic life.

ABSTRACT

Due to the development of digital technologies, new social issues relating to malicious attacks and unauthorized tampering to speech, such as content replacement and voice morphing, have arisen. Using advanced speech analysis and synthesis tools enables the speech to be tampered without leaving any perceptual clues. As an important information carrier, the originality, integrity, and authenticity of speech signals should be strictly confirmed.

Authentication and tampering detection of digital signals is one of the main applications of digital watermarking. In this thesis, an efficient speech watermarking method is proposed to generate self-embedding speech signals in that hash representation of a speech signal, which is assumed as watermark, is embedded into the signal itself without affecting the original quality. The proposed system is intended to satisfy blindness, inaudibility, and fragility against malicious modifications.

The proposed watermarking method in this thesis is a kind of fragile watermarking and thus the hash information (watermark) in the tampered regions is destroyed when tampering occurs. This feature helps the receiver to detect and localize the tampering regions by comparing the original hash information and the extracted hash from the received speech. The perfect match of the hash information confirms the integrity and originality of the received speech; otherwise it indicates tampering.

In this thesis, performance of the proposed system is tested on 40 read speech files which are International news and Burmese news read by 20 female and 20 male announcers. The proposed method is implemented in MATLAB and fragility against malicious modification is evaluated by applying various kinds of tampering such as compression, zeroing, adding noise, time scaling, and reverberation attacks on the watermarked speech files. Experimental results show that the proposed method is relevant with the main requirements of a good watermarking scheme: inaudibility and blindness in addition to fragility. Therefore, the proposed system is really useful for applications of criminal investigation and digital forensics where the integrity and originality of the speech evidence is extremely important.

TABLE OF CONTENTS

PAGE

ACKNOWLEDGEMENTS……... i

ABSTRACT…….. iii

TABLE OF CONTENTS…… iv

LIST OF FIGURES……..vi

LIST OF TABLES……..viii

LIST OF EQUATIONS……...ix

CHAPTER 1 INTRODUCTION 1

1.1 Background 1

1.2 Speech Watermarking 3

1.2.1 Application of Speech Watermarking 4

1.3 Related Works.....6

1.4 Objectives of the Thesis 7

1.5 Organization of the Thesis 7

CHAPTER 2 BACKGROUND THEORY 8

2.1 Applied Areas and Role of Speech Secrecy 8

2.2 Speech Protection and Cryptography.....9

2.3 Cryptographic Hash Function 11

2.3.1 Secure Hash Algorithm (SHA) 12

2.3.1.1 The SHA-512 Algorithm 14

2.4 Overview of Data Hiding Techniques 20

2.5 Digital Watermarking 21

2.5.1 Digital Watermarking for Speech 21

CHAPTER 3 SYSTEM IMPLEMENTATION 25

3.1 Generalized Tampering Detection Scheme 25

3.2 Implementation of the Proposed System 26

3.2.1 Watermark Generation and Embedding.....26
3.2.2 Watermark Extraction and Tamper Detection 29

CHAPTER 4 RESULT AND DISCUSSION 31

4.1 Experimental Results 31
4.1.1 Performance Evaluation for Inaudibility.....31
4.1.1.1 Signal-to-Noise Ratio (SNR) 31
4.1.1.2 Log Spectrum Distortion (LSD) 33
4.1.1.3 Evaluation Results for Inaudibility.....33
4.1.2 Performance Evaluation for Fragility.....44
4.1.2.1 Bit Detection Rate (BDR).....45
4.1.2.2 Tampering Types and Evaluation Results ...45
4.2 Tampering Localization 60
4.3 Summary 63

CHAPTER 5 CONCLUSION 64

5.1 Further Extension.....64

REFERENCES 66

PUBLICATION 71

LIST OF FIGURES

Figure	Description	Page
1.1	Watermarking as a communication system	4
2.1	Block diagrams of (a) symmetric-key and (b) asymmetric-key cryptography schemes	11
2.2	Overview of SHA-512	15
2.3	Overview of word expansion in SHA-512	17
2.4	Overview of hash computation in SHA-512	19
2.5	An example LSB replacement method	23
3.1	Generalized tampering detection scheme	26
3.2	Self-embedding speech generation framework	27
3.3	LSB replacement method	27
3.4	Watermark extraction and tampering detection framework	30
4.1	Average SNR, LSD, and elapsed time results for inaudibility	43
4.2	The LSD and SNR results of inaudibility for 4LSB embedding	43
4.3	Waveforms of (a) the “news1(f,B).wav” read speech and (b) its corresponding 4LSB watermarked speech	44
4.4	Comparative analysis of different zeroing attacks	46
4.5	Waveform of the tampered speech by zeroing attack	47
4.6	Waveforms of the tampered speeches by noise addition attack	48
4.7	Comparative analysis of different noise addition attacks	49
4.8	Waveforms of the tampered speeches by reverberation attack	51
4.9	Comparative analysis of different reverberation attacks	51
4.10	Comparative analysis of different concatenation attacks	53
4.11	Comparative analysis of time scaling attack (speed up) with different scale factors	54
4.12	Comparative analysis of time scaling attack (speed down) with different scale factors	55
4.13	Waveforms of the tampered speeches by time scaling (speed-up) attack	56

4.14 Waveforms of the tampered speeches by time scaling (speed-down) attack 57
4.15 Comparative analysis of compression attack with G.711 (A-law) ... 59
4.16 Comparative analysis of compression attack with G.711 (μ-law) 59
4.17 Waveforms of the tampered speeches by G.711compression 60
4.18 Tampering localization map for zeroing attack 61
4.19 Tampering localization map for reverberation attack 62
4.20 Tampering localization map for noise addition attack..... 63

LIST OF TABLES

Table	Description	Page
2.1	Specification of secure hash algorithms	13
2.2	Security strength of SHA variants	14
4.1	Speech files used in the experiment	32
4.2	Inaudibility evaluation for 1LSB based on SNR and LSD	34
4.3	Inaudibility evaluation for 2LSB based on SNR and LSD	36
4.4	Inaudibility evaluation for 4LSB based on SNR and LSD	37
4.5	Inaudibility evaluation for 6LSB based on SNR and LSD	39
4.6	Inaudibility evaluation for 8LSB based on SNR and LSD	40
4.7	Fragility of the proposed method against zeroing	46
4.8	Fragility of the proposed method against noise addition	48
4.9	Fragility of the proposed method against reverberation	50
4.10	Fragility of the proposed method against concatenation	52
4.11	Fragility of the proposed method against time scaling (speed up)	54
4.12	Fragility of the proposed method against time scaling (speed down)	55
4.13	Fragility of the proposed method against G.711 compression	58

LIST OF EQUATIONS

Equation	Description	Page
2.1	Word expansion used in SHA-512 calculation	17
2.2	Sigma function 1 used in word expansion for SHA-512	17
2.3	Sigma function 2 used in word expansion for SHA-512	17
2.4	Sum function 1 used in compression for SHA-512	18
2.5	Sum function 2 used in compression for SHA-512	18
2.6	Majority function used in compression for SHA-512	18
2.7	Choice function used in compression for SHA-512	18
3.1	Frame division	28
3.2	Frame size	28
3.3	Bit pattern conversion	28
3.4	Decomposition of LSB and MSB	29
3.5	Watermark generation with SHA-512 at embedding side	29
3.6	Watermark embedding	29
3.7	Watermark generation with SHA-512 at detection side	29
3.8	Watermark extraction	29
4.1	Calculation of the signal-to-noise ratio	32
4.2	Calculation of the log spectral distortion	33
4.3	Calculation of the bit detection rate	45
4.4	Generation of a reverberated signal	50

CHAPTER 1

INTRODUCTION

This thesis aims to develop an efficient speech watermarking method that can be used for tampering detection in transmission of speech signals. This chapter firstly presents the challenges on today multimedia industry, especially the negative impact of technological advances on speech privacy and security. Then, it continues the discussion on the importance of speech watermarking in protecting speech privacy together with the application areas. Finally, the chapter is concluded by presenting the objectives and organization of the thesis.

1.1 Background

Globalization and the Internet are the most valuable resources for information communication and retrievals these days. The emergence of the state-of-the-art digital technologies and communication systems has significantly impacted human life from communications to social interaction. As a result of advanced technologies, digital multimedia such as digital video, image, audio, and speech are unbelievably easy to transmit and share over a communication channel. People can access these massive information in a more convenient and faster way that cannot be achieved prior to digital technologies [30].

However, the advancements of technology do not have only positive effects on our lives, but also it brings some negative impacts. Since digital technologies enable signals to be delivered in a detached manner crossing time and distances, unforeseen and illegal operations such as content replacement can easily tamper the signals being transmitted. Since digital signals with invaluable importance have been widely used for many occasions, security of those signals has been a tremendously important issue to deal with. In the past, protection of digital signals mainly focused on video, audio, and images which contain commercial values. However, protection of speech signals has also drawn much attention these days as speech has widely been used not only in our daily life such as Voice over Internet Protocol (VoIP) communication [57] and mobile but also for more important areas such as governmental and commercial activities, digital forensics, etc.

In face to face communication case, there is no doubt that what the listeners hear is what the speakers want to express. However, nowadays, people can easily record the speech contents with modern electronic devices such as mobile phones and camcorders. Moreover, some specialized speech analysis and synthesis tools are very professional to produce high naturalness and intelligibility of the tampered speech although important information has been changed. For example, speech content (what the speaker is saying) can be tampered by using voice conversion [56], e.g. a word replacement from "YES" to "NO"; individuality of the speaker (who is saying) can be deliberately transformed to that of another speaker by using speech morphing [22].

Speech content and speaker identity play key roles in criminal investigation and digital forensics [8] because speech can record (1) what happened in a certain place and time and (2) the information provided by the victims or the suspects. Thus, any single word change or forged speaker will result in serious problem for judgment. However, in most cases, speech is not immediately used after being recorded. They have to pass through a series of judicial procedures in which different people may involve. Since not everyone involved is trustful and improper actions (intentionally or unintentionally) taken to handle, examine, and store the speech are possible to destroy their originality, it is very difficult to ensure the integrity of speech after complicated processing. To confirm the speech is best suited to the unique acquisition environment and the truth, investigation about whether the speech has been tampered since its creation should be carried out.

As discussed above, on the one hand digital technologies have facilitated greatly the sharing of multimedia signals (especially speech in this thesis), but on the other hand they have also increased the need for protection of the signals from any misuse. Since speech is inevitably used not only in our daily lives but also for more important occasions, speech security and integrity have become an important issue to deal with. In general, there are two categories to provide speech security: active method and passive method. The cryptography [25] as an active method can prevent speech from tampering by setting up a secure delivery of speech from the sender side to the receiver side and speech watermarking as a passive method by means of which speech can be automatically authenticated.

1.2 Speech Watermarking

Security requirements such as data integrity and data authentication can be met by implementing speech watermarking techniques. Speech watermarking is the art of embedding watermark information (e.g. bits, logo etc.), which can permanently exit, into the host speech signal to assure its integrity and origin source authentication without degrading the overall quality and commercial value of the signal itself. The embedded watermark can later be extracted in case it is needed to confirm the ownership of the signal or to check the integrity of the signal.

A speech watermarking system can be modeled as a communication system in which the watermark embedding process is considered as the signal transmitter [19]. The watermark can be seen as the signal to be transmitted and the host can be seen as the noise. Any tampering to the watermarked signal (e.g. compression, noising, channel distortion, etc.) can be modeled as the communication channel, which may be any kind of wire or wireless channel such as mobile communication channels, radio broadcasting, and the Internet. Detection of the embedded watermark corresponds to the detection of signal with the presence of the noise at the receiver. Figure 1.1 shows a schematic overview of the watermarking system configuration.

Speech watermarking techniques should generally satisfy the requirements of inaudibility and blindness. Inaudibility means that embedding of watermarks should be inaudible to human auditory system, i.e. it should not affect the sound quality of the original speech. Blindness indicates the watermarks that will later be used for tampering detection or copyright confirmation should be extracted without referring the host signal. Watermark extraction should naturally be blind for practicality.

Additionally, speech watermarking can also be categorized into robust and fragile watermarking [30] based on the application areas. Robust watermarking means that watermarks should resist against moderate tampering or normal signal processing operations. Robust watermarking is mainly used for copyright protection. Fragile watermarking means that watermarks should be sensitive to tampering and easy to be destroyed once tampering has been made to the watermarked signal [10]. Fragile watermarking enables speech to be authenticated in a more suitable and durable way and thus it is mainly used for authentication and tampering detection of speech.

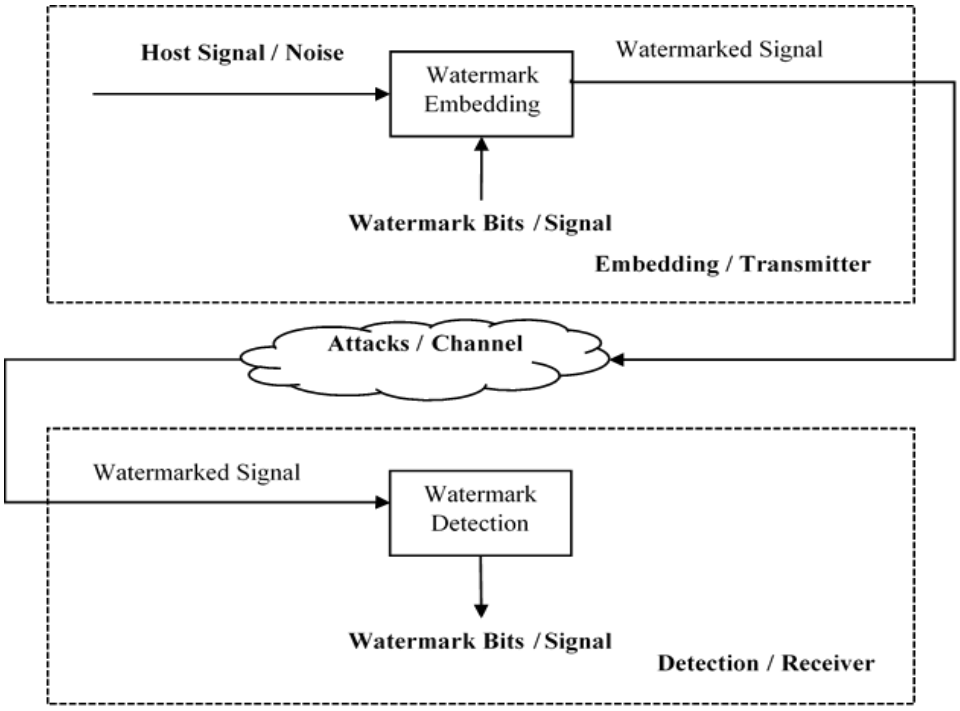


Figure 1.1: Watermarking as a communication system

Speech tampering detection schemes are basically split to two main categories: i) schemes that just verify the originality of speech without localizing the tampering and ii) schemes that can localize the tampering regions. Tampering localization is critical because knowledge of where the signal has been altered can be effectively used to indicate the valid region of the signal, to infer the motive and the possible adversaries. Moreover, the type of alteration may be determined from the knowledge of tampering localization.

In this thesis, the proposed system realizes an efficient tampering detection and localization method for speech signals. It is developed based on a combined approach of fragile watermarking with hash function to detect unauthorized speech tampering as well as the negative influence that they may cause.

1.2.1 Application of Speech Watermarking

This section discusses how widely speech watermarking techniques are employed in real world applications.

Air traffic control: Air traffic control relies on the voice communication between aircraft pilots and air traffic control operators. It is desirable to transmit aircraft identification data

over the very high frequency (VHF) analog voice channel. If not otherwise explicitly specified (i.e. standard situation), every voice message of aircraft pilot is inherently addressed to the air traffic controller (addressee). The pilot starts the message with his call-sign to identify himself as the addresser of the message. For a safe communication, the correct identification of addresser and addressee is crucial. Incorrect identification can lead to wrong orders, which can have fatal consequences. This "misidentification" is most likely to happen, when aircrafts with similar call-sign present at the same channel and either one addresser or addressee mistakes the two call-signs. This potential risk is usually referred to as call-sign confusion.

Digital watermarking was identified as a possible solution for the problem of call-sign confusion. The technique embeds the addresser's identification as a small digital tag into the voice signal. It is inherently transmitted with the voice signal and can be read by the addressee and then communicated to the control tower for the safety regards [19].

Crime investigation and digital forensics: Consider a case where the police receive a surveillance speech file that has been tampered. If the speech file is authenticated with a traditional signature, the police would assume that the whole speech file is inauthentic and cannot be trusted. If they used a watermark for authentication, they may distinguish reliable parts of the speech from tampered parts. In this way, it would be strong evidence that the identity of someone involved in the crime was removed from the tampered parts [47] [29].

Broadcast monitoring: There are several types of organizations and individuals interested in broadcast monitoring. Advertisers, of course, want to ensure that they receive the exact air time purchased from broadcasting firms. Musicians and actors want to ensure that they receive accurate royalty payments for broadcasts of their performances. Copyright owners also want to ensure that their property is not illegally rebroadcasted by pirate stations. For this broadcast monitoring purpose, watermarking techniques can be utilized in which a unique watermark is embedded in each video or sound clip prior to broadcast. Automated monitoring stations can then receive broadcasts and look for these watermarks to identify when and where each clip appears [11] [54].

In addition to the above mentioned applications, speech watermarking can also be used for ensuring authenticity and integrity of speech signals.

1.3 Related Works

In the last few years, several fragile watermarking techniques have been proposed for authentication and tampering detection of speech signals.

Sarreshtedari et al. [48] proposed a self-recovery speech watermarking method for tampering detection. At the transmitter side, compressed version of a speech signal was generated by a speech codec and protected against tampering by proper channel coding. The channel-coded signal was then embedded in the original speech signal and transmitted. At the receiver side, the channel decoder could recover the compressed speech bitstream from the survived channel code bits. Experimental results showed that the system could recover the tampered segments with proper speech quality for high tampering rates.

Wang et al. [49] proposed a tampering detection scheme for speech signals based on formant enhancement-based watermarking. Watermarks were embedded as slight enhancement of the formants by symmetrically controlling linear spectral frequencies (LSFs) of the corresponding formants. The core idea is to provide inaudibility by taking advantage that humans are not sensitive to slight enhancement of formant. The proposed system ensured robustness against meaningful processing and fragility against tampering.

Unoki and Miyauchi [35] introduced an inaudible watermarking method for detection of tampering in speech signals by employing the characteristics of cochlear delay. Watermarks were embedded by enhancing the phase of the original speech with respect to two kinds of group delays. That system could detect not only the tampering with the content but also that with the speaker individuality and the non-linguistic information.

Celik et al. [32] proposed a watermarking method by introducing small changes to pitch (fundamental frequency) via quantization index modulation (QIM). Insensitivity of human perception to natural variability of pitch enabled the method to be inaudible. The stability of pitch under low data rate compression (e.g. Global System for Mobile communications coder 6.10 and Adaptive Multi-Rate coder) also made the method effective for semi-fragile authentication. Nonetheless, the method had not been designed to be robust against attacks that aimed to obstruct detection of watermarks. For example, a systematic modification of pitch such as re-embedding would typically disable the watermarks.

This thesis also proposes an efficient tampering detection and localization method for speech signals. The proposed system implements a self-embedding speech watermarking method in which hash representation of a speech signal is embedded into the signal itself. As the channel attacks and malicious tampering, various kind of tampering such as compression, zeroing etc. are injected into the watermarked signal and it was found out that the embedded watermark is fragile enough against those attacks. By detecting the embedded watermark in the tampered signal, the system can localize the tampered regions perfectly.

1.4 Objectives of the Thesis

The objectives of the proposed system are as follow.

- To study cryptographic hash functions and speech watermarking techniques
- To develop a speech watermarking method that not only achieves tampering localization but also satisfies inaudibility, blindness, and fragility
- To detect malicious tampering on the transmitted speech signal
- To locate the tampering regions in the tampered speech signal
- To reduce the cost and time needed for retransmission of the entire speech signal by locating the tampered regions, in case the signal was tampered

1.5 Organization of the Thesis

This thesis consists of five main chapters. Chapter 1 introduces the concept of speech watermarking and its application areas. Chapter 2 discusses the theoretical background of the proposed method in detail. Chapter 3 presents the proposed system design: watermark embedding and tampering detection and localization methods in detail. Then, Chapter 4 evaluates the performance of the proposed method based on inaudibility, blindness, and fragility requirements. Finally, Chapter 5 concludes this thesis by discussing the benefits and limitation of the proposed system.

CHAPTER 2

BACKGROUND THEORY

As mentioned in Chapter 1, unauthorized tampering in speech signals has brought serious problems when verifying their originality and integrity. Digital watermarking can effectively check if the speech signals have been tampered by embedding digital data into them and this thesis implements such a method. This chapter discusses the theories behind speech watermarking and cryptographic hashing techniques, which are the backbone theories of this thesis.

2.1 Applied Areas and Role of Speech Secrecy

Speech is the most natural tool by means of which people can communicate with others to express one's thoughts, emotions, and willingness. With the help of digital technologies these days, speech signals have been widely used starting from our daily life till government, military, and commercial activities. Therefore, the protection of speech signals has drawn much attention. The following applications highlight the importance of speech secrecy.

VoIP communications: Voice over Internet Protocol (VoIP) refers to making telephone calls over the IP network [39] [46]. The technique of VoIP is becoming increasingly popular as people can make phone calls at reduced expenses over the Internet rather than the company's network, just like email systems. VoIP is available on phones, computers, and other devices, and not only a way to transport data but also a foundation for more enriched multimedia communication applications with speech and video. For most business applications, the VoIP calls are managed with private networks so that the information security can be ensured. As for common customs, since the VoIP calls connect directly to the Internet, attackers can steal speech data, record conversations, or spy on the calls. Therefore, there exists a potential threat that the captured speech data may be misused for crime issues. Motivated by this, effective measures should be used to protect the speech data transmitted via VoIP communication.

Digital forensics: In digital forensics, speech signals are usually employed as a kind of digital evidence [43]. The speech evidences may record the criminal activities or

interrogation of the suspects and victims. These evidences need to be recovered from electronic/digital devices and then submitted to the court to support or oppose a hypothesis. As the judicial proceedings are largely based on these evidences, the integrity and originality of digital speech evidences should be strictly confirmed. If there is tampering conducted by malicious intends to mislead the listeners, e.g. cutting or adding some key words in speech sentences or transforming the individuality of a speaker to that of another speaker, unfair results will come out and people will question the fairness of the court and lose the confidence of social justice.

Government activities: Every element of the government officers' statement greatly affects the society and human life. Consider the speech recordings involving official secrets were stolen and attacked, e.g. content concatenation or replacement. Once the modified speech recordings are released publicly, it is tough for the government to address the issue and bring it under control [28].

Commercial investigation: Forensics may be used in private section such as business and intrusion investigations [30] [27]. In a corporation, confidentialities such as negotiation, board meeting, and economic decision are usually recorded for emergency needs and treated with extraordinary secrecy. Once the speech data is tampered illegally, it may cause serious economic losses.

2.2 Speech Protection and Cryptography

Cryptography deals with encryption that is defined as protecting the information (speech signal in this case) by converting it into an unrecognizable format. It is generally believed that the cryptography realizes a secure transmission over the untruthful medium. Secure transmission hereby indicates that the speech signal sent from the sender side cannot be accessed or altered by the third party. Only the legal recipient at the receiver side, who has an authorized key, is able to decrypt the speech. It does not hide the existence of the message from the attacker instead it renders the content of the message garbled to unauthorized people.

As a result of the advances in modern computer these days, cryptography has become available to everyone for producing an encrypted signal with complexity that most powerful attack algorithms cannot work out in million years. As an effective security tool,

cryptography meets most of the security interests in the Internet and telecommunication by preventing confidential messages from unauthorized access.

Two main processes of cryptography are (1) encryption that transforms data (plaintext) into an unreadable format (ciphertext) and (2) decryption that restores the unreadable file to its original format. A secure transmission provided by cryptography generally relates to the following three requirements [25]:

(i) **Authentication** refers to proving and guaranteeing the identity, i.e. speech signal is not sent by an impostor instead of the specific sender.

(ii) **Privacy** concerns with ensuring that any attackers cannot access the transmitted speech except the intended receiver.

(iii) **Integrity** indicates that the received speech has not been altered prior to receiver in any way from the original.

Three typical cryptographic schemes that accomplish the above requirements are secret-key (symmetric-key) scheme, public-key (asymmetric-key) scheme, and hash function based scheme. In the secret-key scheme, both the sender and receiver use the same single key to encrypt and decrypt the digital signal. In the public-key scheme, two keys are required: the public key is usually known to everybody and the private key is only known to the indented receiver. These two keys are generated in a way that it is impossible to work out the private key based on the public key. The block diagrams of these two schemes are shown in Figure 2.1.

Conventional cryptography-based methods can also be used to prevent speech signals from tampering, in which only legal recipients will be provided with a key to decrypt the speech. However, the key in cryptography attaches great importance to the security of the whole system. If the decryption key is captured by an illegal user or if the system used to encrypt the signal is intruded by hackers, cryptography cannot protect the signals anymore. Thus, there exists a defect in cryptographic techniques that once the decrypted speech is edited or distributed, or if the decryption key is captured by illegal user, cryptography cannot provide any information to track the speech for its originality and integrity. In order to solve that defect, cryptographic hash functions have come in useful [2] [7].

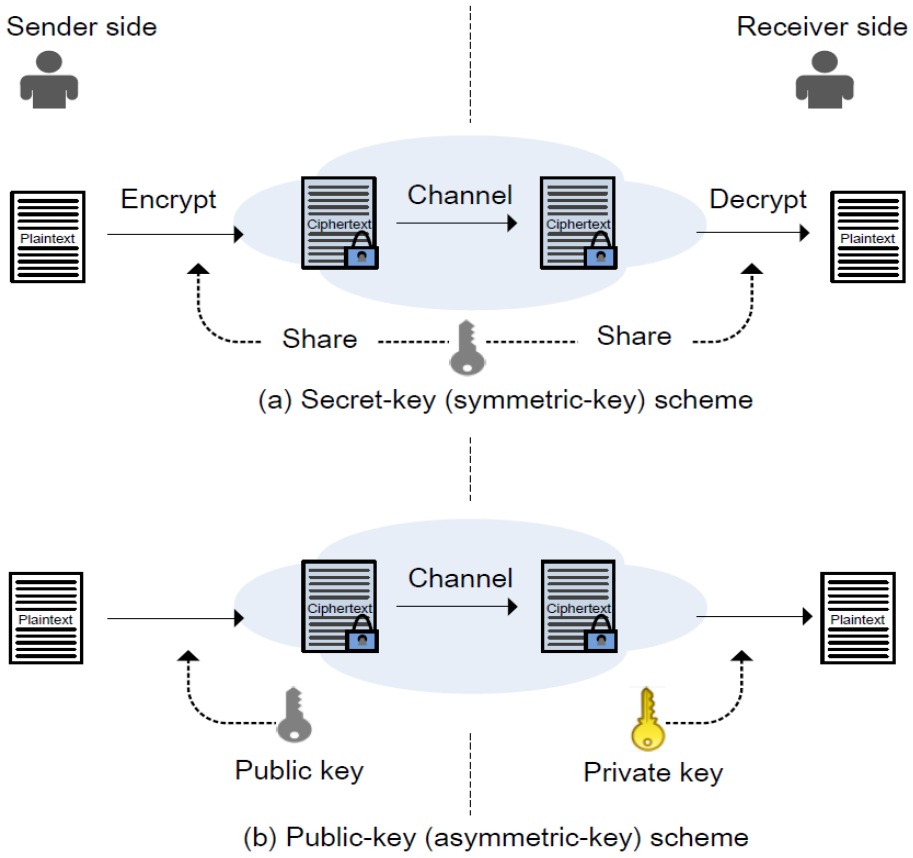


Figure 2.1: Block diagrams of (a) symmetric-key and (b) asymmetric-key cryptography schemes

2.3 Cryptographic Hash Function

A standard cryptographic hash function is a one-way encryption in which it takes a message of arbitrary length and creates a message digest (hash value) of fixed length. It has the property that minimal alterations of input data significantly change the resulting hash value, aka avalanche effect. It allows one to easily verify that some input data maps to a given hash value; but even if the hash value is known, it is deliberately difficult to reconstruct the input data. Hash algorithms are typically used to provide a digital fingerprint of a file's contents and often used to ensure that the file has not been altered by an intruder or virus.

Secure hash functions should satisfy the properties of pre-image resistant, second-image resistant, and collision resistant [2].

Pre-image resistant: Given hash value *h*, it should be hard to find message *m* such that *h* = hash(*m*) (there is no inverse function);

Second-image resistant: Given message $m1$, it should be hard to find message $m2$ ($\neq m1$) such that hash($m1$) = hash($m2$);

Collision-resistant: It should be hard to find messages $m1$ and $m2$ ($m2 \neq m1$) such that hash($m1$) = hash($m2$);

There have been many cryptographic hash functions published and one of them is Message-Digest (MD) functions. Among the MD family, MD4 and MD5 are quite well-known. The numerals refer to the functions being the fourth and fifth designs from the same hash-function family. In 1990, MD4 was first proposed by Ron Rivest and MD5 followed shortly thereafter as its stronger version. Their design had great influence on subsequent construction of hash functions.

The MD5 [41] was first published as an Informational RFC (Request for Comments). It is a message digest algorithm that takes a message of arbitrary length as an input and produces a 128-bit "message digest" of the input as an output. Since that time, the MD5 had been extensively studied and new cryptographic attacks had been discovered. The published attacks against MD5 show that it is not prudent to use MD5 when collision resistance is required [41] [42].

Following the structure of MD4 and MD5, Secure Hash Algorithm (SHA) is a cryptographic hash function designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS) [51]. Secure one-way hash functions are recurring tools in cryptosystems like the symmetric block ciphers. They are highly flexible primitives that can be used to obtain authenticity, privacy, and integrity.

2.3.1 Secure Hash Algorithm (SHA)

There are three SHA algorithms which are structured differently and distinguished as SHA-0, SHA-1, and SHA-2 respectively. The hash function SHA-0 is 160 bits long and was published in 1993 as a federal standard by the NIST [51]. However, it was withdrawn by the NSA shortly after publication due to cryptographic weaknesses and superseded by the revised version SHA-1 published in 1995 [52].

On the other hand, SHA-2 differs significantly from its predecessor, SHA-1. It was published in 2002 as a U.S. federal standard by the NIST [53]. The SHA-2 family consists

of six hash functions that yield digests (hash values) with length of 224, 256, 384, or 512 bits respectively. Based on the digest length, they are differently named as SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256.

All SHA algorithms are iterative one-way hash functions that can provide a condensed fixed-length representation known as message digest of an input message. These algorithms enable the determination of message's integrity: any changes to the message will, with a very high probability, result in a different message digest. Table 2.1 summarizes the basic properties of these hash algorithms.

Each SHA algorithm can be described in two stages: preprocessing and hash computation. Preprocessing involves padding a message, parsing the padded message into *m*-bit blocks, and setting the initialization vector (IV) and round constants to be used in the hash computation. Values and sizes of the IV and additive constants may differ depending on the algorithm. Hash computation generates a message schedule from the padded message which is then used along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by hash computation is used to determine the message digest.

The SHA algorithms differ most significantly in security strengths that are provided for the data being hashed. Security strength of a hash algorithm depends on the size of the output hash code. For an *n*-bit hash, its security strength is $2^{(n/2)}$ bits. Thus, security strength of the MD4 and MD5 algorithms with 128-bit hash is no more than 64 bits. Security strengths of the whole SHA family are shown in Table 2.2.

Table 2.1: Specification of secure hash algorithms

Algorithm		**Message Size (bits)**	**Block Size (bits)**	**Word Size (bits)**	**Rounds**	**Output Size (bits)**
SHA-0		$<2^{64}$	512	32	80	160
SHA-1						
SHA- 2	SHA-224	$<2^{64}$	512	32	64	224
	SHA-256	$<2^{64}$	512	32	64	256
	SHA-384	$<2^{128}$	1024	64	80	384
	SHA-512	$<2^{128}$	1024	64	80	512
	SHA-512/224	$<2^{128}$	1024	64	80	224
	SHA-512/256	$<2^{128}$	1024	64	80	256

Table 2.2: Security strength of SHA variants

Algorithm		Output Size (bits) (*n*)	Security Bits (*n*/2)
SHA-0		160	<34 (collision found)
SHA-1		160	<63 (collision found)
SHA-2	SHA-224	224	112
	SHA-256	256	128
	SHA-384	384	129
	SHA-512	512	256
	SHA-512/224	224	112
	SHA-512/256	256	128

The NIST has announced that the security of SHA-256, SHA-384, and SHA-512 matches the security of the Advanced Encryption Standard (AES) with complexity of the best attack as 2^{128}, 2^{192}, and 2^{256}, respectively. According to the security strengths shown in Table 2.2, the SHA-512 provides better security level with 256 security bits rather than the other family members.

In addition, the SHA-512 consists of 80 steps of operation, known as rounds. Usually, more rounds imply more security and hence harder to break. Thus, the SHA-512 is applied in the proposed system in this thesis to implement a secure self-embedding watermarking scheme. More detail of the SHA-512 is discussed below.

2.3.1.1 The SHA-512 Algorithm

As discussed above, SHA-512 is a family of SHA-2 and used to hash a message $\boldsymbol{M}$ with a length of l bits, where $0 \leq \boldsymbol{l} < 2^{128}$ (the empty message has length 0). If $\boldsymbol{l}$ is a multiple of 8, it can be represented in hex for compactness. The output size is 512 bits and overview of the SHA-512 is shown in Figure 2.2. Like the other hash algorithms, SHA-512 consists of preprocessing and hash computation stages.

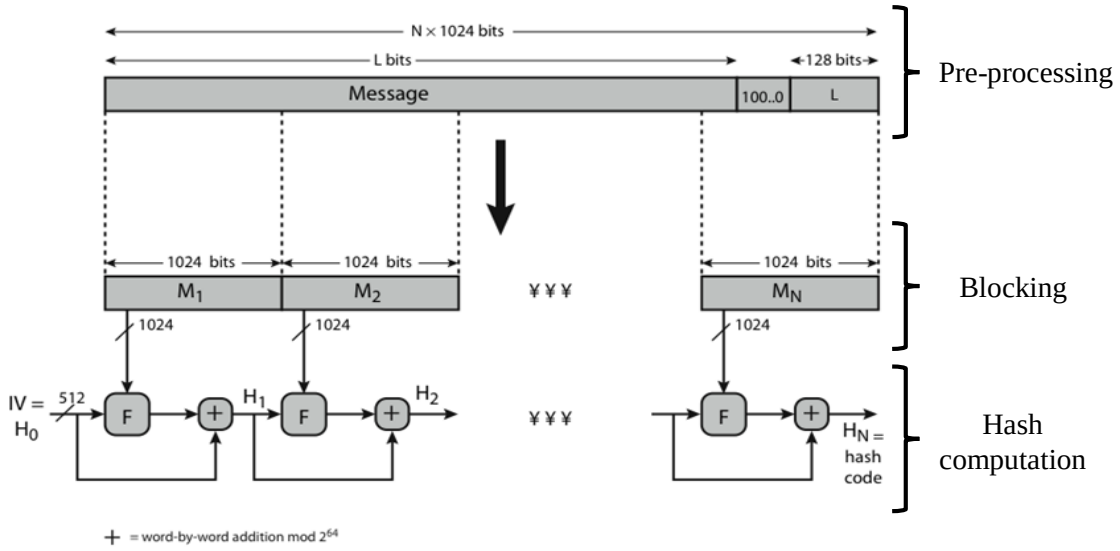


Figure 2.2: Overview of SHA-512

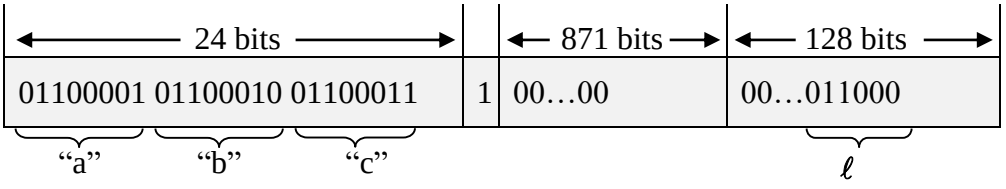
(1) Preprocessing

Padding the message ***M***, parsing the padded message into message blocks, and setting the initial hash values and round constants are the preprocessing steps.

Step 1: Padding the Message

The purpose of padding is to ensure that the padded message is a multiple of the block size of SHA-512 (i.e. 1024 bits). Padding can be done on a message before hash computation begins, or at any other time during the hash computation prior to processing the block(s).

As shown in Figure 2.2, if ***M*** was not a multiple of 1024 bits, it is padded with the bit "1" followed by ***k*** zero bits and then padded with ***l*** in 128-bit binary at the end, where ***k*** is the smallest, non-negative solution to the equation $l+1+k \equiv 896 \bmod 1024$. The reason why the bit "1" is firstly padded is that otherwise collisions occur between the padded message and the original message ended with zeros. For example, an 8-bit ASCII message "abc" has length $l = 8\times3 = 24$ bits and thus, $k = 871$ zero bits are padded as follows.



Step 2: Parsing the Message

The padded message is divided into N 1024-bit blocks, M_1, M_2, …, M_N. Each 1024-bit block is expressed as sixteen 64-bit words, labeled as W_0, W_1, …, W_{15}. Then, those blocks are sequentially processed when computing the message digest.

Step 3: Initialize the Hash Values and Round Constants

Before the hash computation begins, set eighty 64-bit round constants, K_0, K_1, …, K_{79}. These round constants are expressed in hex format and they represent the first 64 bits of the fractional parts of the cube roots of the first 80 prime numbers (2 … 409).

K[0…79] :=

0x428a2f98d728ae22, 0x7137449123ef65cd, 0xb5c0fbcfec4d3b2f, 0xe9b5dba58189dbbc,
0x3956c25bf348b538, 0x59f111f1b605d019, 0x923f82a4af194f9b, 0xab1c5ed5da6d8118,
0xd807aa98a3030242, 0x12835b0145706fbe, 0x243185be4ee4b28c, 0x550c7dc3d5ffb4e2,
0x72be5d74f27b896f, 0x80deb1fe3b1696b1, 0x9bdc06a725c71235, 0xc19bf174cf692694,
0xe49b69c19ef14ad2, 0xefbe4786384f25e3, 0x0fc19dc68b8cd5b5, 0x240ca1cc77ac9c65,
0x2de92c6f592b0275, 0x4a7484aa6ea6e483, 0x5cb0a9dcbd41fbd4, 0x76f988da831153b5,
0x983e5152ee66dfab, 0xa831c66d2db43210, 0xb00327c898fb213f, 0xbf597fc7beef0ee4,
0xc6e00bf33da88fc2, 0xd5a79147930aa725, 0x06ca6351e003826f, 0x142929670a0e6e70,
0x27b70a8546d22ffc, 0x2e1b21385c26c926, 0x4d2c6dfc5ac42aed, 0x53380d139d95b3df,
0x650a73548baf63de, 0x766a0abb3c77b2a8, 0x81c2c92e47edaee6, 0x92722c851482353b,
0xa2bfe8a14cf10364, 0xa81a664bbc423001, 0xc24b8b70d0f89791, 0xc76c51a30654be30,
0xd192e819d6ef5218, 0xd69906245565a910, 0xf40e35855771202a, 0x106aa07032bbd1b8,
0x19a4c116b8d2d0c8, 0x1e376c085141ab53, 0x2748774cdf8eeb99, 0x34b0bcb5e19b48a8,
0x391c0cb3c5c95a63, 0x4ed8aa4ae3418acb, 0x5b9cca4f7763e373, 0x682e6ff3d6b2b8a3,
0x78a5636f43172f60, 0x84c87814a1f0ab72, 0x8cc702081a6439ec, 0x90befffa23631e28,
0xa4506cebde82bde9, 0xbef9a3f7b2c67915, 0xc67178f2e372532b, 0xca273eceea26619c,
0xd186b8c721c0c207, 0xeada7dd6cde0eb1e, 0xf57d4f7fee6ed178, 0x06f067aa72176fba,
0x0a637dc5a2c898a6, 0x113f9804bef90dae, 0x1b710b35131c471b, 0x28db77f523047d84,
0x32caab7b40c72493, 0x3c9ebe0a15c9bebc, 0x431d67c49c100d4c, 0x4cc5d4becb3e42b6,
0x597f299cfc657e2a, 0x5fcb6fab3ad6faec, 0x6c44198c4a475817

Next, the algorithm uses eight initial 64-bit hash values, H_0, ..., H_7, each expressed as hex format. These hash values are obtained by taking the first 64 bits of the fractional parts of the square roots of the first 8 prime numbers (2 … 19).

H_0 := 0x6a09e667f3bcc908	H_1 := 0xbb67ae8584caa73b
H_2 := 0x3c6ef372fe94f82b	H_3 := 0xa54ff53a5f1d36f1
H_4 := 0x510e527fade682d1	H_5 := 0x9b05688c2b3e6c1f
H_6 := 0x1f83d9abfb41bd6b	H_7 := 0x5be0cd19137e2179

(2) Hash computation

The hash computation stage consists of four steps: preparing the message schedule, setting the initial working variables, compression function, and computing the hash value. The final hash value generated by this stage is used to determine the message digest.

Step 1: Preparing the message schedule

To achieve better security, the sigma functions ($\sigma 0$ and $\sigma 1$) expands the initial 1024-bit message block M expressed as sixteen 64-bit words W_i, where $0 \leq i \leq 15$, to eighty 64-bit words W_i with $0 \leq i \leq 79$. Overview of the message schedule process is shown in Figure 2.3.

$$W_i = \begin{cases} W_i, & 0 \leq i \leq 15 \\ W_{i-16} + \sigma_0 + W_{i-7} + \sigma_1, & 16 \leq i \leq 79 \end{cases} \quad (2.1)$$

with

σ_0 = (W_{i-15} rightrotate 1) xor (W_{i-15} rightrotate 8) xor (W_{i-15} rightshift 7) (2.2)

σ_1 = (W_{i-2} rightrotate 19) xor (W_{i-2} rightrotate 61) xor (W_{i-2} rightshift 6) (2.3)

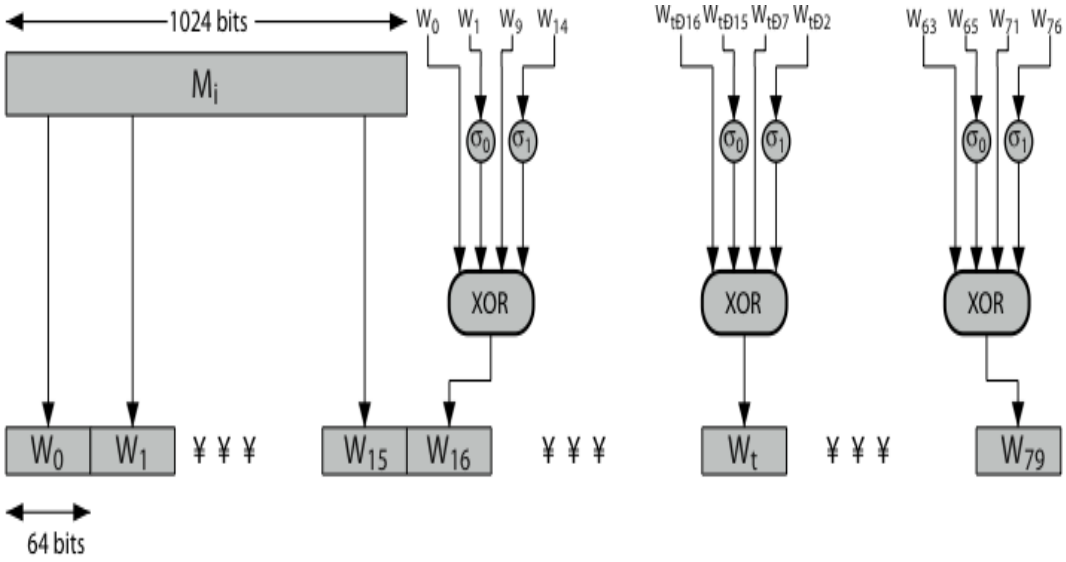


Figure 2.3: Overview of word expansion in SHA-512

Step 2: Setting the initial working variables

Assign the current hash values $\boldsymbol{H}_0$, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{H}_3$, $\boldsymbol{H}_4$, $\boldsymbol{H}_5$, $\boldsymbol{H}_6$, $\boldsymbol{H}_7$ to the eight 64-bit working variables ***a***, ***b***, ***c***, ***d***, ***e***, ***f***, ***g***, and ***h***.

$$(a, b, c, d, e, f, g, h) := (H_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7)$$

Step 3: Compression function

The 80 rounds of compression function, which is made up of three logical functions (sum (Σ), majority, and choice), are applied on the working variables as follows. A round of the compression function is shown in Figure 2.4.

For ***i*** from 0 to 79

{

$\boldsymbol{t}1 := \boldsymbol{h} + \Sigma(\boldsymbol{e}) + \mathrm{Ch}(\boldsymbol{e},\boldsymbol{f},\boldsymbol{g}) + \boldsymbol{K_i} + \boldsymbol{W_i}$

$\boldsymbol{t}2 := \Sigma(\boldsymbol{a}) + \mathrm{Maj}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$

$\boldsymbol{h} := \boldsymbol{g}, \boldsymbol{g} := \boldsymbol{f}, \boldsymbol{f} := \boldsymbol{e}, \boldsymbol{e} := \boldsymbol{d} + \boldsymbol{t}1, \boldsymbol{d} := \boldsymbol{c}, \boldsymbol{c} := \boldsymbol{b}, \boldsymbol{b} := \boldsymbol{a}, \boldsymbol{a} := \boldsymbol{t}1 + \boldsymbol{t}2$

}

$\Sigma(\boldsymbol{a})$ = (***a*** rightrotate 28) xor (***a*** rightrotate 34) xor (***a*** rightrotate 39) (2.4)

$\Sigma(\boldsymbol{e})$ = (***e*** rightrotate 14) xor (***e*** rightrotate 18) xor (***e*** rightrotate 41) (2.5)

$\mathrm{Maj}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{c})$ = (***a*** and ***b***) xor (***a*** and ***c***) xor (***b*** and ***c***) (2.6)

$\mathrm{Ch}(\boldsymbol{e},\boldsymbol{f},\boldsymbol{g})$ = (***e*** and ***f***) xor ((not ***e***) and ***g***) (2.7)

$\boldsymbol{K_i}$ = eighty 64-bit additive constant

$\boldsymbol{W_i}$ = eighty 64-bit word derived from the expanded input block

Step 4: Computing the hash value

Finally, the output of the compression function is added to the initialized hashes or hash values of the previous round to give the new intermediate hash values, according to the Davies-Meyer construction. After repeating the steps 1-4 of hash computation ***N*** times (i.e. after processing all message blocks), the final hash values $\boldsymbol{H}_0$ to $\boldsymbol{H}_7$ is the 512-bit message digest of ***M***.

$$\boldsymbol{H}_0 := \boldsymbol{H}_0 + \boldsymbol{a}; \quad \boldsymbol{H}_1 := \boldsymbol{H}_1 + \boldsymbol{b}; \quad \boldsymbol{H}_2 := \boldsymbol{H}_2 + \boldsymbol{c}; \quad \boldsymbol{H}_3 := \boldsymbol{H}_3 + \boldsymbol{d}$$
$$\boldsymbol{H}_4 := \boldsymbol{H}_4 + \boldsymbol{e}; \quad \boldsymbol{H}_5 := \boldsymbol{H}_5 + \boldsymbol{f}; \quad \boldsymbol{H}_6 := \boldsymbol{H}_6 + \boldsymbol{g}; \quad \boldsymbol{H}_7 := \boldsymbol{H}_7 + \boldsymbol{h}$$

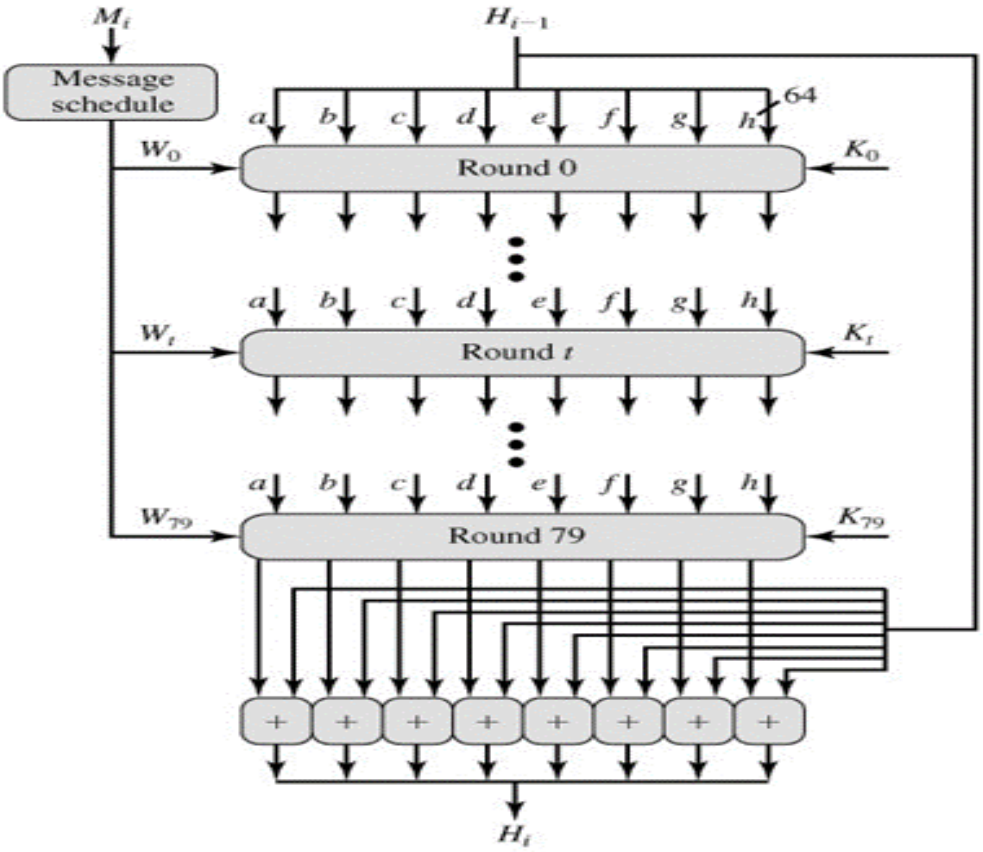


Figure 2.4: Overview of hash computation in SHA-512

Example hashes of the SHA-512

As an example, the hash values generated by the SHA-512 algorithm for two messages are shown below. Even though the messages differ only in a single character (the first message does not have a full stop), it can be seen that the resulting hash values are completely different.

SHA512("The quick brown fox jumps over the lazy dog")
=ae547d9586f6a73f73fbac0435ed76951218fb7d0c8d788a309d785436bbb642e93a252a9
54f23912547d1e8a35ed6e1bfd7097821233fa0538f3db854fee6

SHA512("The quick brown fox jumps over the lazy dog.")
=91ea1245f20d46ae9a037a989f54f1f790f0a47607eeb8a14d12890cea77a1bbc6c7ed9cf2
05e67b7f2b8fd4c7dfd3a7a8617e45f3c463d481c7e586c39ac1ed

2.4 Overview of Data Hiding Techniques

Data and information hiding techniques have been proposed as a complement to cryptography [58] [1]. As indicated by the name, these techniques hide or embed additional information (e.g. digital data/message, logo, identification marks, and serial number) in the digital signals [33] [9]. Since information hiding just adds additional information to the digital data, it does not prevent the data from being accessed and used. Compared with cryptography, data hiding techniques concentrate on hiding information for particular purposes rather than securing the communications [45].

Two main branches in information hiding can be found in the literature: one is steganography [4] and the other is digital watermarking [40] [50]. Steganography refers to embedding confidential information such as message, image, audio, speech, or video within another digital signal (cover signal) without attracting suspicion in such a way that the existence of the information is undetectable. The cover signal may be an image, audio, text file, or even video. Stego key is also required for embedding and extracting the secret message.

Digital watermarking is the art of embedding digital information (e.g. copyright notice) into the host signal of text, image, audio, speech, or video with no perceptibility of the existence of the additional information. The embedded data is generally referred as watermarks. Those watermarks should be extractable and must resist against intentional and unintentional attacks or be fragile against malicious modification (tampering). Both steganography and watermarking take advantages of the redundant components in digital signals to hide data.

Comparatively speaking, steganography focuses on hiding the existence of embedded information from being discovered by third party during communication, whereas watermarking aims to protect the digital signal with embedded information and detect unauthorized tampering. Therefore, digital watermarking has been widely applied not only for digital signal protection but also for tampering detection.

In this thesis, a digital speech watermarking technique is implemented to detect tampering and authenticate the originality of speech signals.

2.5 Digital Watermarking

Compared with cryptography, digital watermarking does not prevent users from accessing the signal. Moreover, since watermarks are embedded directly within the signal, the embedded information can permanently exist and is difficult to be removed. Thus, watermarking enables signal to be protected in a more suitable and durable way.

Digital watermarking was originally motivated when signals with massive commercial values became available in digital form [39] [15] [14]. In general, the copy of digital signal is completely the same as the original one. Watermarking has great concern to music manufacturers and publishing companies since unauthorized copies usually lead to huge commercial loss. Digital watermarking has been found in many applications, such as copyright protection, authentication, broadcast monitoring, digital forensics, secure communication, crime investigation, and so on.

Watermarking technique during the evolution was used on images and termed as image watermarking. In image watermarking, text or any another image is embedded into host image for ownership protection. While it is getting mature, more attention has been paid to extensively used audio and speech [50] [3]. In the field of audio watermarking, watermarks are embedded within the audio products in the form of serial numbers or identification codes for several purposes such as recording the copyright ownership, identifying the producers, tracing the distributions, etc. In this case, inaudibility and robustness are controlled as the top priority since inaudibility keeps the commercial value of the audio product intact and robustness guarantees a reliable extraction of the copyright information after the distribution process. However, since inaudibility and robustness conflict with each other, watermarking that can satisfy both inaudibility and robustness are difficult to be realized.

2.5.1 Digital Watermarking for Speech

With the increase in mobile and the Internet communication, speech signals are often used for information transmission. Speech watermarking technique is generally used to conceal the secret messages. However, that secret speech signal is transmitted using untrusted media and an eavesdropper might realize that secret communication has taken place. Speech watermarking can deal with this issue by addressing two questions: one is whether

the speech is original and the other is whether the speech has been tampered since its creation.

An effective audio/speech watermarking technique should satisfy the basic requirements of inaudibility, robustness or fragility, and blindness. Inaudibility means that the embedding of watermarks should not degrade the sound quality of signal for its applications. Robustness means that allowable processing and common attacks to a watermarked signal (e.g. re-sampling and re-quantization) should not destroy the embedded watermarks. Fragility means that the embedded watermarks should be sensitive to tampering and thus easy to be destroyed once tampering has been made to the watermarked signal. Blindness means that watermarks should be detected without referring to any information of the original signal [35] [36].

Based on the application areas, watermarking techniques can be classified into robust and fragile watermarking.

1. Robust watermarking techniques are used in applications where the protection of copyright information is required. Robust watermarks must resist against intentional or unintentional channel attacks. They are used to identify and ensure the legitimate ownership, to track and prevent illegal copying, and to solve the copyright violation problems.
2. Fragile watermarking techniques are used in applications in which data integrity and detection of unauthorized tampering are the major concerns.

Based on the intended application area, some watermarking techniques should satisfy robustness, whereas others should satisfy fragility.

In the literature, watermarking techniques related to image and video have been studied rigorously [16] [13]. Digital watermarking for audio/speech, however, is more challenging since the human auditory system (HAS) [55] is more sensitive in comparison with the human visual system (HVS) due to its wide dynamic range. Nonetheless, relatively successful watermarking algorithms regarding to audio/speech signals have been proposed in the literature and applied in some real situations effectively. These algorithms can be divided into time-domain and transformation-domain methods.

Speech watermarking in time domain: One of the well-known time-domain speech watermarking methods is the least significant bit (LSB) replacement method [37]. In order

to realize inaudibility, the LSB method takes advantage of the fact that human ear is not sensitive to slight modification to the insignificant bits. It is based on the substitution of the LSBs of the carrier signal with the bit pattern from the watermark noise and works as follows [38]. Consider a speech signal with real-valued samples. Each sample value is converted into an integer hence it will be easy to replace the bits. If the sample value is 138 (10000110 in binary) and the watermark bit is 1, the value of the watermarked sample will be 10000111 (139 in decimal).

More than one LSB can be used to embed the watermark data, depending on which inaudibility and robustness may vary. The more LSBs are used to encode the watermark, the more audible the embedding effect as noise and the more robust the method is. Figure 2.5 shows an example of the LSB replacement method in which two LSBs of a 16-bit signal are replaced with the watermark bit 01.

Another time-domain based speech watermarking method is echo-hiding methods [22] [59] in which the mask effects in time domain are utilized to achieve inaudibility. These methods embed data into an original speech signal by introducing an echo in the time domain. Most conventional echo hiding methods had the problem concerning the robustness against malicious attacks and to solve that, the time-spread echo method was then proposed [5]. However, most of the time-domain methods were prone to be not robust. Time domain based speech watermarking methods are simplest to realize and data embedding rate is also very high because watermark embedding is just changing the amplitude of the speech samples, but they are less robust. Therefore, transform domain based watermarking methods have been developed.

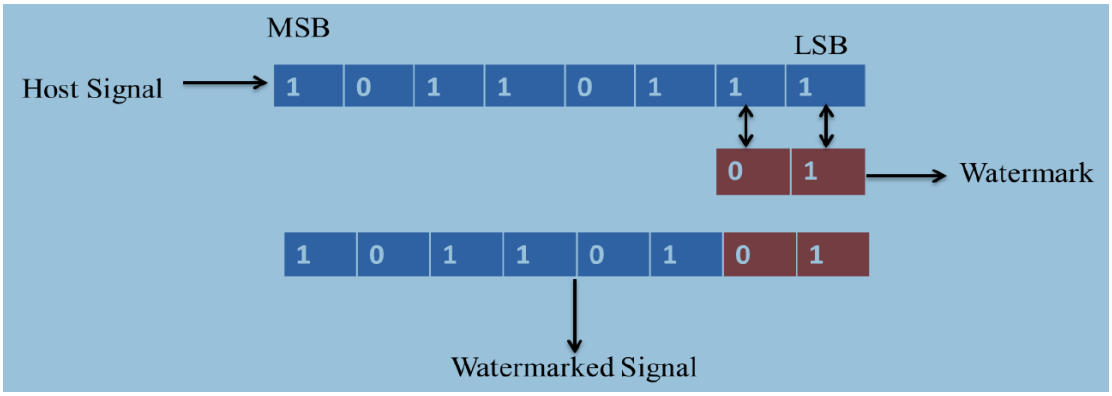


Figure 2.5: An example LSB replacement method

Speech watermarking in transform domain: Speech watermarking techniques were also implemented in transform domain with the purpose of achieving stronger robustness. One of them is spread spectrum techniques [26] [23] in which hidden pseudorandom data are spread throughout the frequency spectrum. Then, the watermark extraction is done by calculating the correlation between the pseudorandom noise data and the watermarked speech signal. Linear spread spectrum techniques apply DSSS/BPSK (direct sequence spread spectrum/binary phase shift keying) to embed confidential data into the host speech signal.

Watermarking based on the Human Auditory System (HAS): Since the HAS is more sensitive than the HVS, more watermarking methods tended to exploit the properties of the HAS and apply such knowledge to obtain the better performance [55]. As per the psychoacoustics, which is the scientific study of sound perception and audiology, human ear can nominally hear sounds in the range 20 Hz to 20 kHz. In addition, there is another interesting characteristic of sound, which is known as auditory masking, in which a softer sound is not heard (masked) in the presence of a louder sound (masker). By exploiting those sound features, HAS based watermarking methods embed watermarks in perceptually inaudible parts of the speech while leaving the sensitive parts intact to realize inaudibility.

Some examples of the HAS based speech watermarking methods are:
(1) a method proposed by Celik et al. by introducing small changes to fundamental frequency [10], (2) a method proposed by Unoki and Hamada, which took the advantages of the characteristics of cochlear delay (CD) in which human is unable to discriminate an enhanced group delay from the original speech [35], and (3) a method proposed by Fallahpour and Megas by utilizing the property of absolute hearing threshold [34].

In summary, speech watermarking methods based on HAS or those developed in transform domain outperform time domain methods in terms of inaudibility and robustness. Thus, those methods are widely used in applications which demand robust watermarking. However, they might increase time and computational complexity.

In this thesis, the main aim is to develop a speech watermarking method which is easily fragile against malicious tampering; no need for strong robustness. Thus, a fragile watermarking method with acceptable inaudibility is developed in time domain.

CHAPTER 3

SYSTEM IMPLEMENTATION

This chapter firstly presents the generalized architecture of a tampering detection scheme for speech signals. Then, it explains the step-by-step implementation of the proposed system, an efficient speech tampering detection and localization method, in detail. This chapter focuses the methodologies mainly used in the proposed system: secure hash function for watermark generation and proposed data hiding method for watermark embedding.

3.1 Generalized Tampering Detection Scheme

As discussed in Chapter 1, watermarking system can be modeled as a communication system in which the watermark embedding process is considered as the signal transmitter and the watermark extraction process is considered as the signal receiver. The watermark can be seen as the signal to be transmitted and any operation to the watermarking (e.g. compression, noising, etc.) can be modeled as the communication channel.

Figure 3.1 shows the generalized architecture of a speech tampering detection scheme for checking whether tampering has occurred to the speech signals during transmission. It consists of three main processes: watermark embedding, extraction, and tampering detection processes. At the embedding side, watermark h_o is embedded into a speech signal $x(n)$ to construct the watermarked signal $y(n)$ which will then be transmitted. At the extraction side, the watermark is blindly extracted from $\hat{y}(n)$ (perhaps the tampered $y(n)$). The extracted watermark $\hat{h}_o$ is then compared with the original watermark h_o to check whether tampering has occurred. If a speech watermarking method satisfies fragility, once tampering occurred, the watermarks in the tampered regions will be destroyed. Therefore, tampering can be detected by the mismatched bits between h_o and $\hat{h}_o$. If there is no mismatch, it confirms the integrity of the received signal.

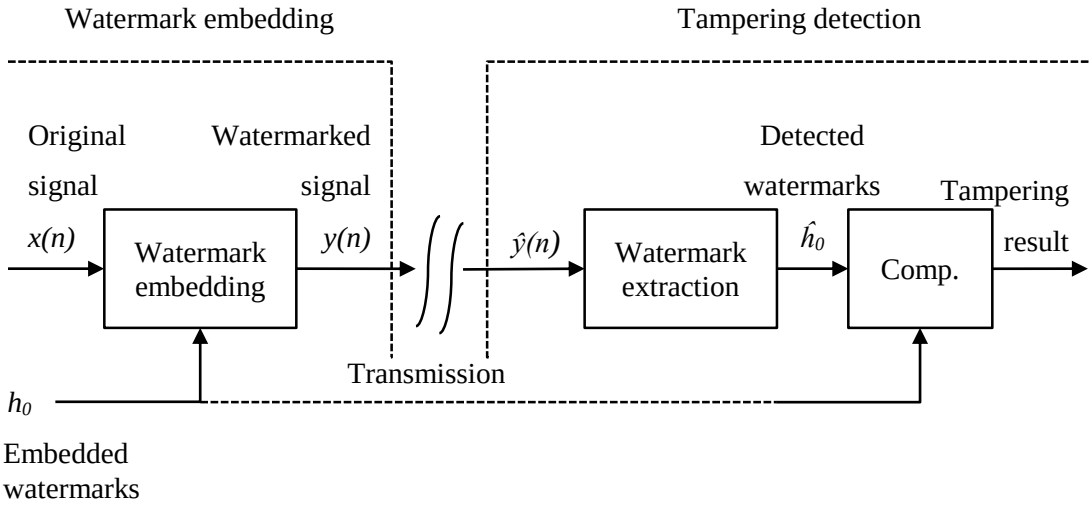


Figure 3.1: Generalized tampering detection scheme

3.2 Implementation of the Proposed System

The proposed system consists of two main parts: (i) watermark generation and embedding, and (ii) watermark extraction and tamper detection.

3.2.1 Watermark Generation and Embedding

Figure 3.2 shows the framework of the self-embedding speech generation process. As shown in the figure, the first step is the "frame decomposition" step in which the input speech is segmented into frames. The size of a frame depends on the size of the watermark to be embedded. In the proposed system, the size of the watermark is 512 bits and each 4-bit will be embedded in each sample of the host speech, and thus the frame size is 128 samples. The reason why each 4-bit of watermark is embedded in each sample will be explained later in Chapter 4.

As the data hiding method in this system, the least significant bit (LSB) replacement method is used due to the effect of replacing the LSB bits with watermarks is less perceptible to the human auditory system and thus provides better inaudibility. As stated by the name, the LSB method basically overwrites the least significant bit of the binary sequence of each sample in a host speech signal with binary equivalent of the watermark bit. Figure 3.3 shows how the LSB method works.

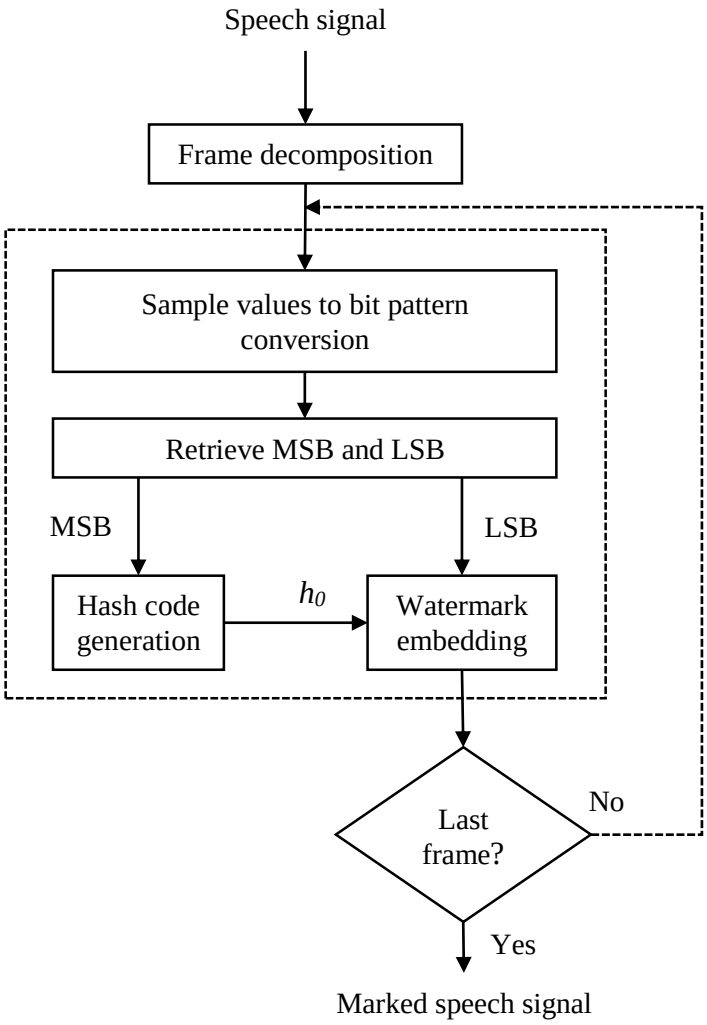


Figure 3.2: Self-embedding speech generation framework

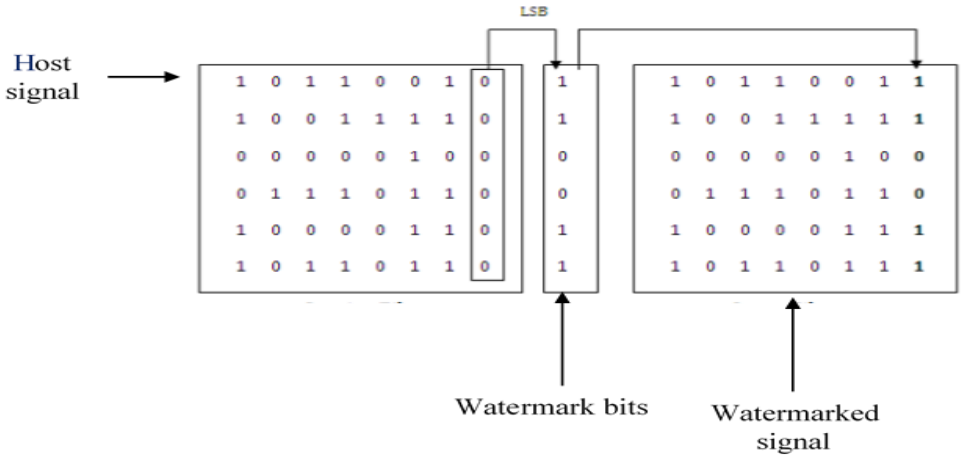


Figure 3.3: LSB replacement method

LSB coding [37] [38] is one of the earliest techniques studied in the information hiding and watermarking area of digital speech signal (as well as other media types). It is because the speech signals have real values as samples hence it will be easy to replace the bits. For example, if the sample value is 138, then its binary equivalent is 10000110. If the watermark bit of 1 is to be embedded, the value of the sample will be 10000111 in binary which is 139 in decimal. If the watermark bits of 00 is to be embedded, the value of the sample will be10000100 in binary which is 136 in decimal. More than one LSB can be used to carry the watarmark bits as per the system requirement, but at the expense of reduced inaudability.

The watermark used in this system is the hash generated from the host speech itself, thus referred to as self-embedding. The choice of the watermark type depends on the system requirement. If the system aims for copyright protection of speeches, the watermark should be something that identifies the owner, such as logo, sign, etc. If the system is intended for tampering detection, self-embedding watermark such as hash of the host speech is one of the commonly used ones.

To generate the watermark in this system, the secure hash algorithm (SHA) is used for the sake of improved authentication. The SHA-512 accepts any message of arbitrary length and generates a 512-bit hash code. As discussed in Chapter 3, although there are many well-known hashing algorithms, the SHA-512 is considered very secure and no attacks are known presently. Unlike the previous SHA versions, the SHA-512 uses different additive constants, shift amounts, and 80 rounds of hash computation process for stronger security [53].

The following is the step-by-step procedure of the proposed watermark generation and embedding process. Consider a 16-bit, 8-kHz sampled speech signal $\boldsymbol{S}$.

Step 1: The $\boldsymbol{S}$ is divided into $\boldsymbol{N}$ frames, each with size of 128 samples.

$$\boldsymbol{S} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \ldots, \boldsymbol{f}_N\}, \tag{3.1}$$

$$\boldsymbol{f}_i = \{\boldsymbol{s}_{(1,i)}, \boldsymbol{s}_{(2,i)}, \boldsymbol{s}_{(3,i)}, \ldots, \boldsymbol{s}_{(128,i)}\}, \tag{3.2}$$

where $\boldsymbol{i} = \{1, \ldots, N\}$.

Step 2: For a frame, each sample value is converted into its 16-bit equivalent.

$$\boldsymbol{s}_j = \{\boldsymbol{b}_{(15,j)}, \boldsymbol{b}_{(14,j)}, \boldsymbol{b}_{(13,j)}, \ldots, \boldsymbol{b}_{(0,j)}\}, \tag{3.3}$$

where $\boldsymbol{j} = \{1, \ldots, 128\}$.

Step 3: Each bit pattern is separated into the most significant bit (MSB) and LSB. Out of the 16 bits representing each sample, $n_w = 4$ LSB are dedicated to the watermark embedding, while the rest $n_m = (16 - n_w)$ MSB are left intact during the embedding process.

$$b_m = \begin{bmatrix} s_1\,(b_{15}, \dots, b_4) \\ \cdot \\ \cdot \\ \cdot \\ s_{128}\,(b_{15} \dots, b_4) \end{bmatrix}, \quad b_w = \begin{bmatrix} s_1\,(b_3, \dots, b_0) \\ \cdot \\ \cdot \\ \cdot \\ s_{128}\,(b_3, \dots, b_0) \end{bmatrix}, \tag{3.4}$$

where $b_m = n_m \times 128 = 12 \times 128 = 1536$ MSB bits of the whole frame and $b_w = 4 \times 128 = 512$ LSB bits of the whole frame that carry the watermark.

Step 4: The SHA-512 algorithm is used to generate the hash bits from b_m.

$$h_o = \mathbf{SHA512}(b_m), \tag{3.5}$$

where h_o is the original hash data (512 bits).

Step 5: The b_w of a frame is replaced with the hash bits h_o of that frame.

$$f' = \mathbf{Embed}(b_w, h_o). \tag{3.6}$$

The above steps 2 to 5 are repeated for all frames and finally, the watermarked speech signal S' is produced.

3.2.2 Watermark Extraction and Tamper Detection

Figure 3.4 shows the framework of the watermark extraction and tampering detection process at the receiver side. The watermark extraction process is performed blindly on the received speech signal S' (may be tampered or not tampered). The followings are the step-by-step procedure of the extraction process.

Step 1-3 of the watermark embedding process is applied again on the received signal S' to retrieve the MSB bits b'_m and the LSB bits b'_w of each frame.

Step 4: The hash bits are then generated from b'_m.

$$\hat{h_o} = \mathbf{SHA512}(b'_m), \tag{3.7}$$

where $\hat{h_o}$ is the generated watermark from the MSB part.

Step 5: Extract the watermark from b'_w.

$$\hat{h_e} = b'_w, \tag{3.8}$$

where $\hat{h_e}$ is the extracted watermark from the LSB part.

The above steps 2 to 5 must be repeated for all frames. For tampering detection, the extracted hash data $\hat{h}_e$ are compared to the generated hash data $\hat{h}_o$ of the same frame. The speech frames are marked as healthy for matching hash data, and otherwise tampered.

By tracking the frame index of the tampered frames, the tampering regions on the speech signal can be localized. By localizing the tampering regions, it can be determined which part of the speech is needed to be retransmitted. It is a great help for applications that need the whole healthy speech signal, e.g. medical report transmission and military message passing, as it can reduce the cost and time needed for retransmission of the entire signal.

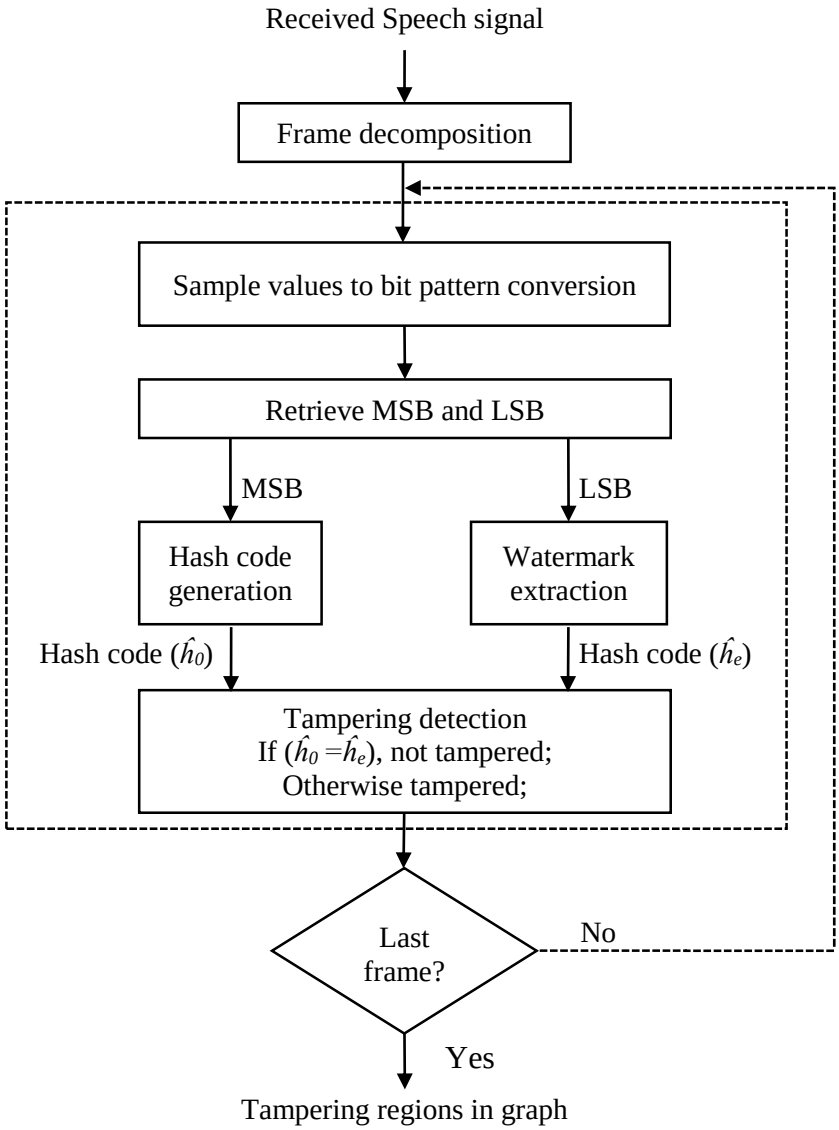


Figure 3.4: Watermark extraction and tampering detection framework

CHAPTER 4

RESULTS AND DISCUSSION

For an efficient tampering detection scheme, its top priority is to be able to detect tampering and locate tampering regions correctly in case the signal was tampered during transmission. This chapter evaluates and analyzes the performance of the proposed tampering detection method by means of inaudibility, blindness, and fragility. Evaluations are done based on the three matrices: signal-to-noise ratio (SNR) and log spectral distortion (LSD) for inaudibility, and bit detection rate (BDR) for fragility.

4.1 Experimental Results

This section analyzes the performance of the proposed system by using 40 test speech files (male/female Burmese and English read speech). Each is a 16-bit, 8 kHz sampled WAVE file. Table 4.1 shows the description of the speech files, where *B* and *E* mean Burmese and English read speech, *f* and *m* mean female and male speaker respectively. There is no specific reason behind choosing spoken language of speech files and gender of speakers; it is just for rich variety of test speech. Those choices also have no effect on the performance of the system.

4.1.1 Performance Evaluation for Inaudibility

Inaudibility means that the noise introduced by embedding the watermark information into the host speech signal does not affect the speech quality and thus the quality degradation is not detectable by the human auditory system. In order to verify inaudibility, SNR and LSD measures are used. These measures can estimate the degradation between the original and the watermarked speech signals.

4.1.1.1 Signal-to-Noise Ratio (SNR)

The SNR is used to know the amount by which the speech signal is corrupted by the noise. It is defined as the ratio of the summed squared magnitude of the clean signal

Table 4.1: Speech files used in the experiment

Sr No.	Signal	Length (sec)	No. of Samples	Sr No.	Signal	Length (sec)	No. of Samples
1	news1 (f,B)	7	60416	21	news21(m,E)	10	80396
2	news2 (f,E)	7	60505	22	news22(f,E)	10	80640
3	news3 (m,B)	7	61520	23	news23(f,B)	10	81628
4	news4 (m,B)	7	61792	24	news24(m,B)	10	81766
5	news5 (f,E)	7	62310	25	news25(m,E)	10	81894
6	news6 (f,B)	7	62976	26	news26(m,B)	10	81923
7	news7 (m,B)	7	63466	27	news27(f,E)	10	83240
8	news8 (m,B)	8	64011	28	news28(f,B)	10	84224
9	news9 (f,B)	8	65156	29	news29(m,E)	10	84889
10	news10(m,B)	8	66504	30	news30(f,E)	10	84924
11	news11(m,B)	8	66659	31	news31(f,E)	10	86156
12	news12(f,B)	8	69376	32	news32(m,E)	11	88575
13	news13(m,E)	8	70491	33	news33(f,E)	11	88774
14	news14(m,E)	8	70759	34	news34(m,E)	11	88878
15	news15(m,E)	9	73371	35	news35(f,E)	11	89014
16	news16(f,E)	9	73737	36	news36(f,B)	11	89344
17	news17(m,E)	9	74532	37	news37(f,B)	11	90752
18	news18(f,B)	9	76037	38	news38(m,E)	11	92032
19	news19(m,B)	9	76672	39	news39(m,B)	11	92357
20	news20(f,E)	9	77301	40	news40(f,B)	11	95729

$\boldsymbol{s(n)}$ to the summed squared magnitude of the noise signal (difference between the $\boldsymbol{s(n)}$ and the watermarked speech signal $\hat{\mathbf{s}}\boldsymbol{(n)}$ in this case). The SNR in dB is calculated as follows.

$$SNR = 10 * \log_{10} \frac{\sum_{n=1}^{N} s(n)^2}{\sum_{n=1}^{N} \{s(n) - \hat{s}(n)\}^2} \quad [dB], \tag{4.1}$$

where $\boldsymbol{n}$ is the sample index and $\boldsymbol{N}$ is the total number of samples in the speech signal. As per the International Federation of Phonographic Industry (IFPI) [24], the SNR ≥ 40 dB means good inaudibility, i.e. the noise introduced by the watermark embedding process does not affect the speech quality. The higher the SNR means the better the speech quality.

4.1.1.2 Log Spectral Distortion (LSD)

The LSD defined in Eq. (4.2) is a frequency-domain measure and used to compute the spectral distance between the original speech signal and the watermarked speech signal.

$$LSD = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(10 * log_{10}\frac{|Y(w,m)|^2}{|W(w,m)|^2}\right)^2} \text{ [dB]}, \quad (4.2)$$

where $\boldsymbol{m}$ is the frame index, $\boldsymbol{M}$ is the total number of frames, and $\boldsymbol{Y}(\boldsymbol{w},\boldsymbol{m})$ and $\boldsymbol{W}(\boldsymbol{w},\boldsymbol{m})$ are the spectra of $\boldsymbol{m}$-th frame in the original and watermarked speech signals, respectively. LSD of 1.0 dB is chose as the criterion, and a lower value indicates a less distortion [51].

4.1.1.3 Evaluation Results for Inaudibility

As discussed in Chapter 3, inaudibility depends on the number of LSB bits used to carry the watermark information. The more the LSBs used for embedding, the higher the data embedding rate (embedded bits per second) but the lower the inaudibility is. In this system, different sets of experiments were carried out to determine how many LSB bits should be used for watermark embedding to achieve a good balance between inaudibility and data embedding rate.

Table 4.2 shows the inaudibility test results when only one LSB bit (1LSB) is used for watermark embedding. As shown in the table, the average SNR is 68.17 dB, LSD is 0.0006 dB, the data embedding rate is 8569 bps, and time consumption is only 0.0045 sec for 7-11 sec long speeches. As discussed in Chapter 3, the proposed method is carried out on frame-by-frame basis and the frame size in Table 4.2 is 512 samples. It is because the hash generated from each frame by the SHA-512 algorithm is 512 bits. It thus needs 512 samples to carry that watermark (one watermark bit per one LSB of a sample). The same concept goes for the frame sizes of the following experiments.

Table 4.3 shows the results of using the first and second LSB (2LSB) for watermark embedding in which the average SNR is 61.17 dB, LSD is 0.0020 dB, data embedding rate is 17137 bps, and elapsed time is 0.0032 sec. Table 4.4, 4.5, and 4.6 show the results of using 4 LSBs, 6 LSBs, and 8 LSBs, respectively, for watermarking embedding. In all experiments, the SNR results are greater than 20 dB and the LSD are lower than 1 dB for all speech files, even for the case of 8LSB embedding where half of the 16 bits used to

encode each sample was modified. It proves that the proposed watermark embedding process yields good inaudibility results. The data embedding rate is also very high. In addition, it can be evident from the results that the more LSBs are modified by embedding the watermarks, the less inaudible the watermark is.

The time elapsed for watermark embedding is also calculated and shown in the tables. In all experiments, the elapsed time is very short compared to the duration of the host speech signal. It is also very important for real-time applications whose cost depends on the time complexity of the algorithm. Both watermark embedding and extraction processes need to be made as fast as possible with greater efficiency. Some of the possible applications where speed is a constraint are audio streaming and airline traffic monitoring.

Table 4.2: Inaudibility evaluation for 1LSB based on SNR and LSD

Signal (8 kHz, 16-bit)	**No. of Samples**	**Watermark Bits**	**Frame Size (sample)**	**No. of Frames**	**SNR (dB)**	**LSD (dB)**	**Elapsed Time (sec)**
news 1 (f,B)	60416	1LSB	512	118	67.81	0.0005	0.0048
news 2 (f,E)	60505	1LSB	512	118	68.30	0.0005	0.0045
news 3 (m,B)	61520	1LSB	512	120	70.84	0.0005	0.0047
news 4 (m,B)	61792	1LSB	512	120	70.80	0.0004	0.0046
news 5 (f,E)	62310	1LSB	512	121	65.59	0.0005	0.0042
news 6 (f,B)	62976	1LSB	512	123	69.77	0.0006	0.0045
news 7 (m,B)	63466	1LSB	512	123	70.69	0.0003	0.0047
news 8 (m,B)	64011	1LSB	512	125	71.28	0.0007	0.0046
news 9 (f,B)	65156	1LSB	512	127	69.84	0.0005	0.0046
news 10 (m,B)	66504	1LSB	512	129	71.10	0.0009	0.0046
news 11 (m,B)	66659	1LSB	512	130	71.03	0.0006	0.0045
news 12 (f,B)	69376	1LSB	512	135	65.51	0.0009	0.0046
news 13 (m,E)	70491	1LSB	512	137	67.74	0.0004	0.0046
news 14 (m,E)	70759	1LSB	512	138	67.73	0.0004	0.0044
news 15 (m,E)	73371	1LSB	512	143	63.80	0.0009	0.0044
news 16 (f,E)	73737	1LSB	512	144	68.25	0.0006	0.0048
news 17 (m,E)	74532	1LSB	512	145	69.40	0.0003	0.0045

news 18 (f,B)	76037	1LSB	512	148	70.69	0.0006	0.0045
news 19 (m,B)	76672	1LSB	512	149	69.04	0.0003	0.0046
news 20 (f,E)	77301	1LSB	512	150	69.16	0.0003	0.0045
news 21 (m,E)	80396	1LSB	512	157	65.23	0.0007	0.0045
news 22 (f,E)	80640	1LSB	512	157	69.37	0.0004	0.0044
news 23 (f,B)	81628	1LSB	512	159	69.49	0.0007	0.0044
news 24 (m,B)	81766	1LSB	512	159	70.88	0.0008	0.0045
news 25 (m,E)	81894	1LSB	512	159	66.42	0.0006	0.0045
news 26 (m,B)	81923	1LSB	512	160	71.12	0.0004	0.0046
news 27 (f,E)	83240	1LSB	512	162	65.80	0.0008	0.0045
news 28 (f,B)	84224	1LSB	512	164	66.74	0.0006	0.0045
news 29 (m,E)	84889	1LSB	512	165	65.07	0.0007	0.0044
news 30 (f,E)	84924	1LSB	512	165	65.81	0.0007	0.0046
news 31 (f,E)	86156	1LSB	512	168	67.29	0.0006	0.0044
news 32 (m,E)	88575	1LSB	512	172	63.74	0.0009	0.0044
news 33 (f,E)	88774	1LSB	512	173	64.79	0.0007	0.0044
news 34 (m,E)	88878	1LSB	512	173	65.52	0.0005	0.0044
news 35 (f,E)	89014	1LSB	512	173	68.30	0.0003	0.0046
news 36 (f,B)	89344	1LSB	512	174	66.74	0.0005	0.0044
news 37 (f,B)	90752	1LSB	512	177	67.06	0.0010	0.0045
news 38 (m,E)	92032	1LSB	512	179	68.16	0.0005	0.0045
news 39 (m,B)	92357	1LSB	512	180	70.56	0.0006	0.0045
news 40 (f,B)	95729	1LSB	512	186	70.44	0.0008	0.0045
Average	**x=77118**	**z=1**			**68.17**	**0.0006**	**0.0045**
Data Embedding Rate (bps) = (x/y[1])*z					**8569**		

[1] Average duration of test speeches = 9 sec

Table 4.3: Inaudibility evaluation for 2LSB based on SNR and LSD

Signal (8 kHz, 16-bit)	No. of Samples	Watermark Bits	Frame Size (sample)	No. of Frames	SNR (dB)	LSD (dB)	Elapsed Time (sec)
news 1 (f,B)	60416	2LSB	256	236	60.82	0.0022	0.0034
news 2 (f,E)	60505	2LSB	256	246	61.31	0.0017	0.0032
news 3 (m,B)	61520	2LSB	256	252	63.86	0.0016	0.0033
news 4 (m,B)	61792	2LSB	256	254	63.86	0.0021	0.0032
news 5 (f,E)	62310	2LSB	256	258	58.52	0.0017	0.0033
news 6 (f,B)	62976	2LSB	256	258	62.76	0.0021	0.0032
news 7 (m,B)	63466	2LSB	256	269	63.70	0.0019	0.0032
news 8 (m,B)	64011	2LSB	256	271	64.30	0.0016	0.0033
news 9 (f,B)	65156	2LSB	256	271	64.08	0.0022	0.0032
news 10 (m,B)	66504	2LSB	256	282	62.82	0.0030	0.0032
news 11 (m,B)	66659	2LSB	256	284	64.01	0.0021	0.0032
news 12 (f,B)	69376	2LSB	256	289	58.50	0.0025	0.0033
news 13 (m,E)	70491	2LSB	256	291	60.76	0.0013	0.0032
news 14 (m,E)	70759	2LSB	256	292	60.72	0.0015	0.0032
news 15 (m,E)	73371	2LSB	256	294	56.83	0.0025	0.0032
news 16 (f,E)	73737	2LSB	256	296	61.19	0.0015	0.0032
news 17 (m,E)	74532	2LSB	256	298	62.38	0.0013	0.0032
news 18 (f,B)	76037	2LSB	256	319	63.67	0.0022	0.0032
news 19 (m,B)	76672	2LSB	256	321	62.04	0.0016	0.0032
news 20 (f,E)	77301	2LSB	256	324	62.17	0.0015	0.0032
news 21 (m,E)	80396	2LSB	256	329	58.25	0.0022	0.0032
news 22 (f,E)	80640	2LSB	256	329	62.34	0.0012	0.0032
news 23 (f,B)	81628	2LSB	256	329	62.46	0.0021	0.0032
news 24 (m,B)	81766	2LSB	256	336	63.90	0.0028	0.0032
news 25 (m,E)	81894	2LSB	256	336	59.43	0.0022	0.0032
news 26 (m,B)	81923	2LSB	256	339	64.15	0.0017	0.0032
news 27 (f,E)	83240	2LSB	256	342	58.77	0.0026	0.0032
news 28 (f,B)	84224	2LSB	256	342	59.77	0.0027	0.0032

news 29 (m,E)	84889	2LSB	256	344	58.08	0.0020	0.0032
news 30 (f,E)	84924	2LSB	256	349	58.80	0.0018	0.0032
news 31 (f,E)	86156	2LSB	256	351	60.33	0.0024	0.0032
news 32 (m,E)	88575	2LSB	256	354	56.78	0.0030	0.0031
news 33 (f,E)	88774	2LSB	256	356	57.81	0.0024	0.0031
news 34 (m,E)	88878	2LSB	256	359	58.56	0.0017	0.0032
news 35 (f,E)	89014	2LSB	256	359	61.27	0.0012	0.0032
news 36 (f,B)	89344	2LSB	256	367	59.69	0.0016	0.0031
news 37 (f,B)	90752	2LSB	256	368	60.03	0.0030	0.0031
news 38 (m,E)	92032	2LSB	256	374	61.13	0.0015	0.0031
news 39 (m,B)	92357	2LSB	256	375	63.60	0.0021	0.0032
news 40 (f,B)	95729	2LSB	256	384	63.42	0.0015	0.0032
Average	**x=77118**	**z=2**			**61.17**	**0.0020**	**0.0032**
Data Embedding Rate (bps)					**17137**		

Table 4.4: Inaudibility evaluation for 4LSB based on SNR and LSD

Signal (8 kHz, 16-bit)	**No. of Samples**	**Watermark Bits**	**Frame Size (sample)**	**No. of Frames**	**SNR (dB)**	**LSD (dB)**	**Elapsed Time (sec)**
news 1 (f,B)	60416	4LSB	128	472	48.53	0.0119	0.0026
news 2 (f,E)	60505	4LSB	128	472	49.03	0.0099	0.0026
news 3 (m,B)	61520	4LSB	128	480	51.55	0.0145	0.0026
news 4 (m,B)	61792	4LSB	128	482	51.53	0.0142	0.0026
news 5 (f,E)	62310	4LSB	128	486	46.24	0.0103	0.0026
news 6 (f,B)	62976	4LSB	128	492	50.44	0.0162	0.0026
news 7 (m,B)	63466	4LSB	128	495	51.37	0.0126	0.0026
news 8 (m,B)	64011	4LSB	128	500	51.97	0.0135	0.0026
news 9 (f,B)	65156	4LSB	128	509	50.54	0.0179	0.0026
news 10 (m,B)	66504	4LSB	128	519	51.78	0.0133	0.0026
news 11 (m,B)	66659	4LSB	128	520	51.70	0.0129	0.0026
news 12 (f,B)	69376	4LSB	128	542	46.19	0.0179	0.0026
news 13 (m,E)	70491	4LSB	128	550	48.42	0.0084	0.0026

news 14 (m,E)	70759	4LSB	128	552	48.48	0.0078	0.0026
news 15 (m,E)	73371	4LSB	128	573	44.47	0.0155	0.0026
news 16 (f,E)	73737	4LSB	128	576	48.97	0.0099	0.0026
news 17 (m,E)	74532	4LSB	128	582	50.09	0.0069	0.0026
news 18 (f,B)	76037	4LSB	128	594	51.38	0.0177	0.0026
news 19 (m,B)	76672	4LSB	128	599	49.76	0.0103	0.0026
news 20 (f,E)	77301	4LSB	128	603	49.83	0.0092	0.0025
news 21 (m,E)	80396	4LSB	128	628	45.97	0.0145	0.0026
news 22 (f,E)	80640	4LSB	128	630	50.07	0.0095	0.0026
news 23 (f,B)	81628	4LSB	128	637	50.18	0.0150	0.0025
news 24 (m,B)	81766	4LSB	128	638	51.60	0.0165	0.0026
news 25 (m,E)	81894	4LSB	128	639	47.14	0.0112	0.0025
news 26 (m,B)	81923	4LSB	128	640	51.84	0.0138	0.0026
news 27 (f,E)	83240	4LSB	128	650	46.47	0.0149	0.0026
news 28 (f,B)	84224	4LSB	128	658	47.45	0.0156	0.0025
news 29 (m,E)	84889	4LSB	128	663	45.77	0.0115	0.0026
news 30 (f,E)	84924	4LSB	128	663	46.48	0.0118	0.0025
news 31 (f,E)	86156	4LSB	128	673	47.97	0.0137	0.0025
news 32 (m,E)	88575	4LSB	128	691	44.47	0.0182	0.0025
news 33 (f,E)	88774	4LSB	128	693	45.50	0.0145	0.0026
news 34 (m,E)	88878	4LSB	128	694	46.21	0.0110	0.0026
news 35 (f,E)	89014	4LSB	128	695	48.97	0.0070	0.0025
news 36 (f,B)	89344	4LSB	128	698	47.40	0.0196	0.0025
news 37 (f,B)	90752	4LSB	128	709	47.75	0.0195	0.0026
news 38 (m,E)	92032	4LSB	128	719	48.85	0.0097	0.0025
news 39 (m,B)	92357	4LSB	128	721	51.28	0.0107	0.0025
news 40 (f,B)	95729	4LSB	128	747	51.09	0.0106	0.0026
Average	**x=77118**	**z=4**			**48.87**	**0.0131**	**0.0026**
Data Embedding Rate (bps)					**34275**		

Table 4.5: Inaudibility evaluation for 6LSB based on SNR and LSD

Signal (8 kHz, 16-bit)	No. of Samples	Watermark Bits	Frame Size (sample)	No. of Frames	SNR (dB)	LSD (dB)	Elapsed Time (sec)
news 1 (f,B)	60416	6LSB	85	710	36.40	0.2281	0.0025
news 2 (f,E)	60505	6LSB	85	711	36.96	0.0684	0.0023
news 3 (m,B)	61520	6LSB	85	723	39.49	0.1013	0.0024
news 4 (m,B)	61792	6LSB	85	726	39.41	0.1381	0.0023
news 5 (f,E)	62310	6LSB	85	733	34.16	0.0683	0.0023
news 6 (f,B)	62976	6LSB	85	740	38.37	0.1771	0.0023
news 7 (m,B)	63466	6LSB	85	746	39.32	0.1056	0.0023
news 8 (m,B)	64011	6LSB	85	753	39.95	0.0903	0.0023
news 9 (f,B)	65156	6LSB	85	767	38.43	0.1612	0.0023
news 10 (m,B)	66504	6LSB	85	782	39.74	0.1140	0.0023
news 11 (m,B)	66659	6LSB	85	784	39.64	0.1126	0.0023
news 12 (f,B)	69376	6LSB	85	816	34.13	0.1537	0.0023
news 13 (m,E)	70491	6LSB	85	829	36.37	0.0440	0.0023
news 14 (m,E)	70759	6LSB	85	832	36.34	0.0437	0.0023
news 15 (m,E)	73371	6LSB	85	863	32.44	0.1482	0.0023
news 16 (f,E)	73737	6LSB	85	867	36.88	0.0569	0.0023
news 17 (m,E)	74532	6LSB	85	876	38.01	0.0546	0.0023
news 18 (f,B)	76037	6LSB	85	894	39.27	0.2684	0.0023
news 19 (m,B)	76672	6LSB	85	903	37.66	0.1673	0.0023
news 20 (f,E)	77301	6LSB	85	909	37.77	0.0524	0.0023
news 21 (m,E)	80396	6LSB	85	945	33.88	0.1146	0.0023
news 22 (f,E)	80640	6LSB	85	948	38.03	0.0576	0.0023
news 23 (f,B)	81628	6LSB	85	960	38.03	0.1139	0.0023
news 24 (m,B)	81766	6LSB	85	961	39.53	0.1698	0.0023
news 25 (m,E)	81894	6LSB	85	963	35.05	0.0809	0.0023
news 26 (m,B)	81923	6LSB	85	963	39.79	0.0850	0.0023
news 27 (f,E)	83240	6LSB	85	979	34.47	0.1318	0.0023
news 28 (f,B)	84224	6LSB	85	990	35.38	0.1339	0.0023

news 29 (m,E)	84889	6LSB	85	998	33.71	0.0859	0.0023
news 30 (f,E)	84924	6LSB	85	999	34.41	0.0874	0.0024
news 31 (f,E)	86156	6LSB	85	1013	35.89	0.1023	0.0023
news 32 (m,E)	88575	6LSB	85	1042	32.41	0.1368	0.0023
news 33 (f,E)	88774	6LSB	85	1044	33.46	0.0997	0.0023
news 34 (m,E)	88878	6LSB	85	1045	34.13	0.0708	0.0023
news 35 (f,E)	89014	6LSB	85	1047	36.93	0.0421	0.0023
news 36 (f,B)	89344	6LSB	85	1051	35.32	0.2860	0.0023
news 37 (f,B)	90752	6LSB	85	1067	35.67	0.3026	0.0023
news 38 (m,E)	92032	6LSB	85	1082	36.79	0.0528	0.0023
news 39 (m,B)	92357	6LSB	85	1086	39.23	0.1131	0.0023
news 40 (f,B)	95729	6LSB	85	1126	39.01	0.1303	0.0023
Average	**x=77118**	**z=6**			**36.80**	**0.1188**	**0.0023**
Data Embedding Rate (bps)					**51412**		

Table 4.6: Inaudibility evaluation for 8LSB based on SNR and LSD

Signal (8 kHz, 16-bit)	**No. of Samples**	**Watermark Bits**	**Frame Size (sample)**	**No. of Frames**	**SNR (dB)**	**LSD (dB)**	**Elapsed Time (sec)**
news 1 (f,B)	60416	8LSB	64	944	24.09	1.5463	0.0022
news 2 (f,E)	60505	8LSB	64	945	24.88	0.7298	0.0022
news 3 (m,B)	61520	8LSB	64	961	27.44	1.0280	0.0023
news 4 (m,B)	61792	8LSB	64	965	27.41	1.1386	0.0022
news 5 (f,E)	62310	8LSB	64	973	21.92	0.6641	0.0023
news 6 (f,B)	62976	8LSB	64	984	26.15	1.4374	0.0022
news 7 (m,B)	63466	8LSB	64	991	27.26	0.8589	0.0022
news 8 (m,B)	64011	8LSB	64	1000	27.90	0.8619	0.0022
news 9 (f,B)	65156	8LSB	64	1018	26.16	1.1853	0.0022
news 10 (m,B)	66504	8LSB	64	1039	27.68	1.0717	0.0022
news 11 (m,B)	66659	8LSB	64	1041	27.58	0.9871	0.0022
news 12 (f,B)	69376	8LSB	64	1084	22.06	1.4375	0.0022
news 13 (m,E)	70491	8LSB	64	1101	24.33	0.3996	0.0022

news 14 (m,E)	70759	8LSB	64	1105	24.31	0.3791	0.0022
news 15 (m,E)	73371	8LSB	64	1146	20.43	1.2043	0.0022
news 16 (f,E)	73737	8LSB	64	1152	24.85	0.5711	0.0022
news 17 (m,E)	74532	8LSB	64	1164	25.93	0.4922	0.0022
news 18 (f,B)	76037	8LSB	64	1188	26.90	1.5649	0.0022
news 19 (m,B)	76672	8LSB	64	1199	25.40	1.3221	0.0022
news 20 (f,E)	77301	8LSB	64	1207	25.71	0.4777	0.0022
news 21 (m,E)	80396	8LSB	64	1256	21.84	1.0478	0.0022
news 22 (f,E)	80640	8LSB	64	1260	25.97	0.4451	0.0022
news 23 (f,B)	81628	8LSB	64	1275	25.69	1.0682	0.0022
news 24 (m,B)	81766	8LSB	64	1277	27.48	1.3137	0.0023
news 25 (m,E)	81894	8LSB	64	1279	22.98	0.7126	0.0022
news 26 (m,B)	81923	8LSB	64	1280	27.74	0.8693	0.0022
news 27 (f,E)	83240	8LSB	64	1300	22.38	1.2022	0.0022
news 28 (f,B)	84224	8LSB	64	1316	23.29	1.2778	0.0022
news 29 (m,E)	84889	8LSB	64	1326	21.48	0.7605	0.0022
news 30 (f,E)	84924	8LSB	64	1326	22.26	0.8720	0.0022
news 31 (f,E)	86156	8LSB	64	1346	23.62	0.9957	0.0022
news 32 (m,E)	88575	8LSB	64	1383	20.36	1.3347	0.0022
news 33 (f,E)	88774	8LSB	64	1387	21.39	0.9710	0.0022
news 34 (m,E)	88878	8LSB	64	1388	21.97	0.6324	0.0022
news 35 (f,E)	89014	8LSB	64	1390	24.91	0.4033	0.0022
news 36 (f,B)	89344	8LSB	64	1396	23.02	1.5471	0.0022
news 37 (f,B)	90752	8LSB	64	1418	23.49	1.7303	0.0022
news 38 (m,E)	92032	8LSB	64	1438	24.72	0.5484	0.0023
news 39 (m,B)	92357	8LSB	64	1443	27.19	0.9839	0.0022
news 40 (f,B)	95729	8LSB	64	1495	26.68	1.1145	0.0022
Average	**x=80590**	**z=8**			**24.67**	**0.9797**	**0.0022**
Data Embedding Rate (bps)					**68549**		

For clearer illustration, the average SNR, LSD, and elapsed time results for the experiments in Table 4.2-4.6 are comparatively shown in Figure 4.1. As discussed above, the SNR values are decreasing and the LSD values are increasing when the more watermark bits are embedding. It is a sign of decreasing speech quality.

From the above results and discussion, it can be clearly seen that there is a tradeoff between the watermark embedding rate and inaudibility. This tradeoff can be controlled based on the system requirement.

In this proposed system, 4LSBs are chosen for watermark embedding due to the following reasons.

1) Figure 4.2 shows the SNR and LSD results yielded by 4LSB embedding for the 40 read speech files. All the resulting SNR and LSD values satisfy the IFPI criteria (LSD $\leq$ 1.0 dB and SNR $\geq$ 40 dB).

2) The time consumption for 4LSB embedding is also acceptable, averagely 0.0026 sec for 7-11 sec long speech files.

3) The 4LSB embedding does not increase the file size. Although the effect of watermark embedding on file size was not explicitly mentioned, the more the watermark bits are embedding, a bit larger the file size is.

To back up the above choice, Figure 4.3 shows the waveforms of one of the test speech signals “news1(f,B).wav” and its corresponding 4LSB embedding signal, as an example. It can be seen that there is no visually difference between the waveforms and thus means no significant distortion to the original signal.

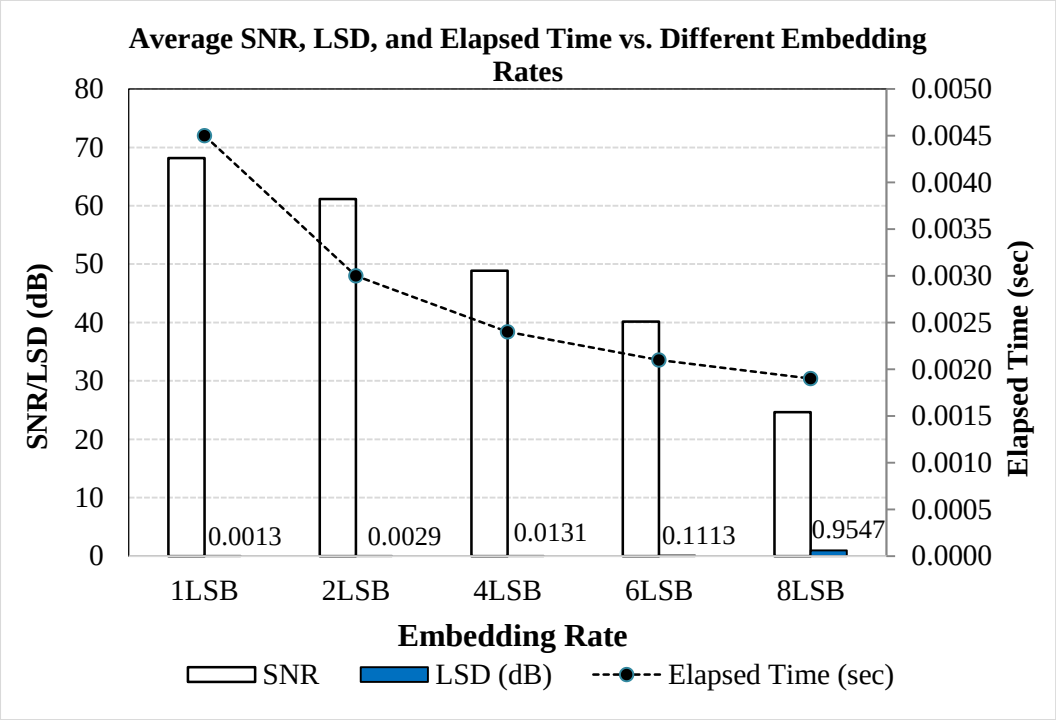


Figure 4.1: Average SNR, LSD, and elapsed time results for inaudibility

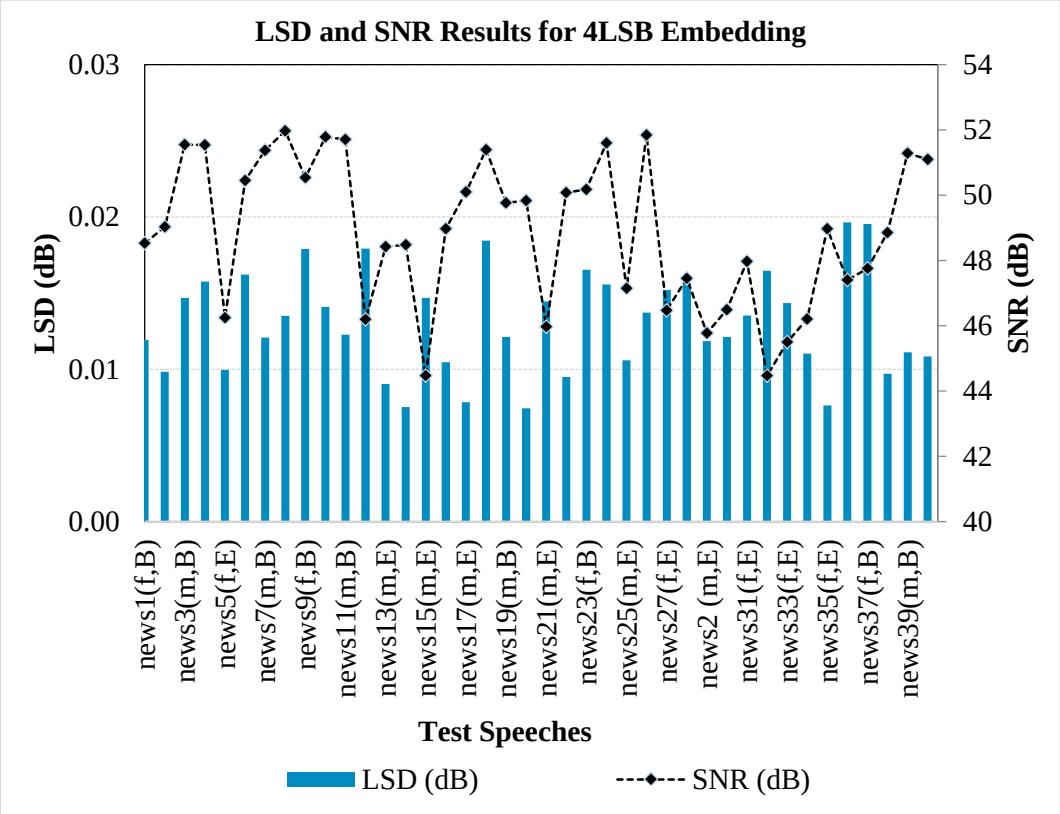


Figure 4.2: The LSD and SNR results of inaudibility for 4LSB embedding

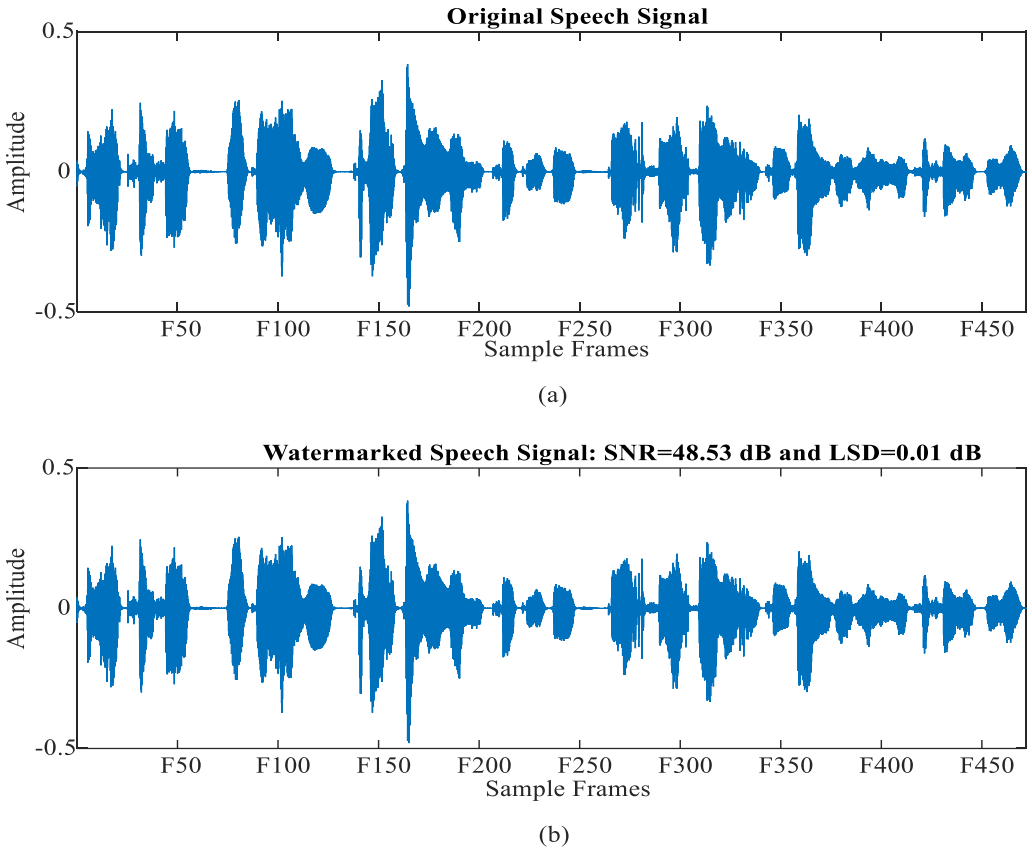


Figure 4.3: Waveforms of (a) the "news1(f,B).wav" read speech and (b) its corresponding 4LSB watermarked speech

4.1.2 Performance Evaluation for Fragility

Fragility means that the embedded watermarks are very sensitive to tampering and easy to be destroyed once tampering has been made. A watermarking method intended to be used for tampering detection should be fragile against several tampering types (e.g., adding noise, signal loss, etc) to authenticate the effectiveness of the embedded watermarks. Many previous works [50] [51] have confirmed the fragility of their methods by carrying out various types of tampering. However, there is no consistent definition for tampering among these works.

In general, tampering is performed based on the motivation of the attackers. In this system, the following tampering types are applied on five test signals, i.e. the signals from Table 4.1 are chosen randomly and watermarked, to test the fragility of the proposed method: zeroing, adding noise, reverberation, concatenation, time scaling, and compression. Fragility is evaluated in terms of the BDR.

4.1.2.1 Bit Detection Rate (BDR)

The BDR is used to measure how similar the extracted watermark is to the original embedded one. BDR is the percentage of the ratio between the correctly extracted watermark bits and the total amount of embedded watermark bits. The BDR can be calculated as follows.

$$\text{BDR} = \frac{\text{M} - \sum_{m=0}^{M-1} \text{s(m)} \oplus \hat{\text{s}}\text{(m)}}{\text{M}} * 100\ (\%)\ , \tag{4.3}$$

where $\boldsymbol{s}(\boldsymbol{m})$ is the embedded watermark, $\hat{s}(\boldsymbol{m})$ is the extracted watermark, and $\boldsymbol{M}$ is the total length of $\boldsymbol{s}(\boldsymbol{m})$. The symbol "$\oplus$" denotes the operation of "exclusive-OR", i.e. if the bit values of $\boldsymbol{s}(\boldsymbol{m})$ and $\hat{s}(\boldsymbol{m})$ are different ($\boldsymbol{s}(\boldsymbol{m}) = 1$ and $\hat{s}(\boldsymbol{m}) = 0$, or $\boldsymbol{s}(\boldsymbol{m}) = 0$ and $\hat{s}(\boldsymbol{m}) = 1$), then "$\boldsymbol{s}(\boldsymbol{m}) \oplus \hat{s}(\boldsymbol{m})$" equals 1; otherwise, "$\boldsymbol{s}(\boldsymbol{m}) \oplus \hat{s}(\boldsymbol{m})$" equals 0. The BDR of 90% is considered as the criterion; a higher BDR indicates the stronger robustness and a lower BDR indicates the stronger confirmation of tampering [51].

4.1.2.2 Tampering Types and Evaluation Results

(1) Zeroing

Replacing the (arbitrary or predefined) samples of a speech signal with zeros introduces the silence in the speech signal. This is similar to the lack of audible sounds or the presence of sounds with very low intensity. Silence in the speech is natural. Speech production process involves generating voiced and unvoiced speech in succession, separated by silence. Silences in speech can be due to hesitation, stutter, self-correction, or a deliberate slowing of speech to clarify or aid the processing of ideas. Without silence between voiced and unvoiced speech, the speech will not be intelligible. Illegal persons may take advantage of this to change a condition in his/her favor. In digital forensics, for example, the words in the speech evidence may be illegally replaced with silence to hide the truth or to remove the identity of someone involved in the crime.

In this system, zeroing attack is carried out by replacing 20%, 40%, 60%, and 80% of the test watermarked speeches with zero. Table 4.7 shows the BDR and SNR results of those experiments. It can be seen from the table that the BDR values are getting smaller when the percentage of zero-replaced samples is increasing. A smaller BDR indicates a stronger confirmation of tampering. The SNR results show how severe the attack is. The

lower SNR indicates the more severe attack. For a clear illustration, the BDR and SNR results for different zeroing attacks are also plotted in Figure 4.4.

Figure 4.5 shows how zeroing attack will look like; straight line parts of the waveform depict the zeroing regions. Hereafter, waveform analysis of the different tampering effects will be carried out on the watermarked "w_news1(f,B).wav" read speech.

Table 4.7: Fragility of the proposed method against zeroing

Signal	Zeroing (%)	20%	40%	60%	80%
w_news1(f,B)	BDR (%)	89.45	78.97	68.40	57.93
	SNR (dB)	6.43	2.16	1.47	0.27
w_news9(f,B)	BDR (%)	89.48	78.95	68.42	57.89
	SNR (dB)	5.58	3.37	1.63	0.50
w_news19(m,B)	BDR (%)	89.47	78.96	68.43	57.90
	SNR (dB)	6.03	3.40	1.76	0.62
w_news22(f,E)	BDR (%)	89.46	78.92	68.46	57.92
	SNR (dB)	5.76	2.87	2.10	1.07
w_news38(m,E)	BDR (%)	89.48	78.96	68.43	57.91
	SNR (dB)	4.55	2.36	1.21	0.74
Average	**BDR (%)**	**89.46**	**78.95**	**68.43**	**57.91**
	SNR (dB)	**5.67**	**2.83**	**1.63**	**0.64**

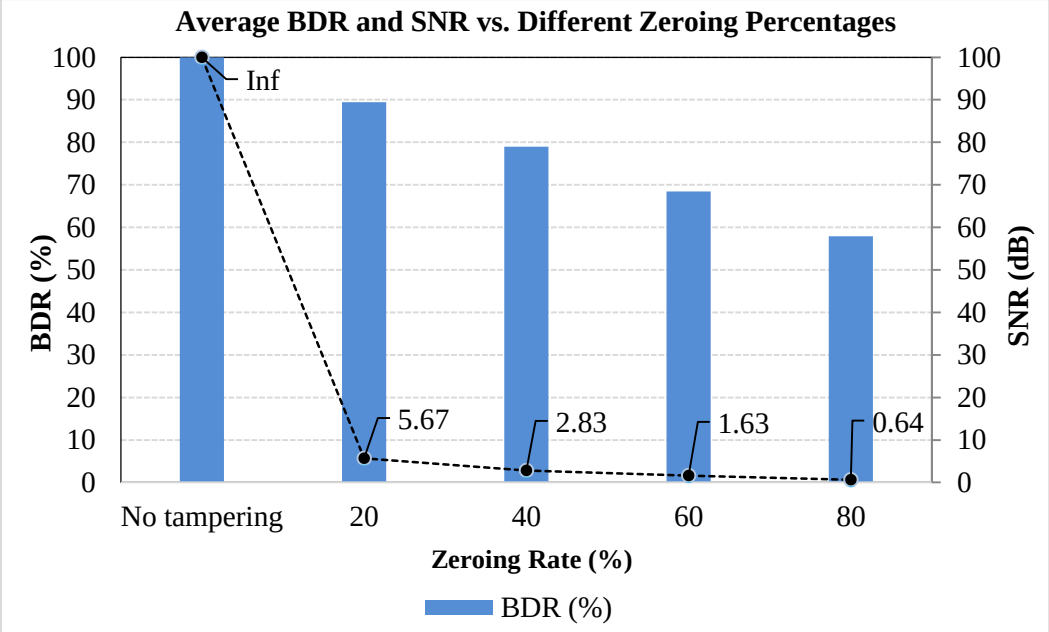


Figure 4.4: Comparative analysis of different zeroing attacks

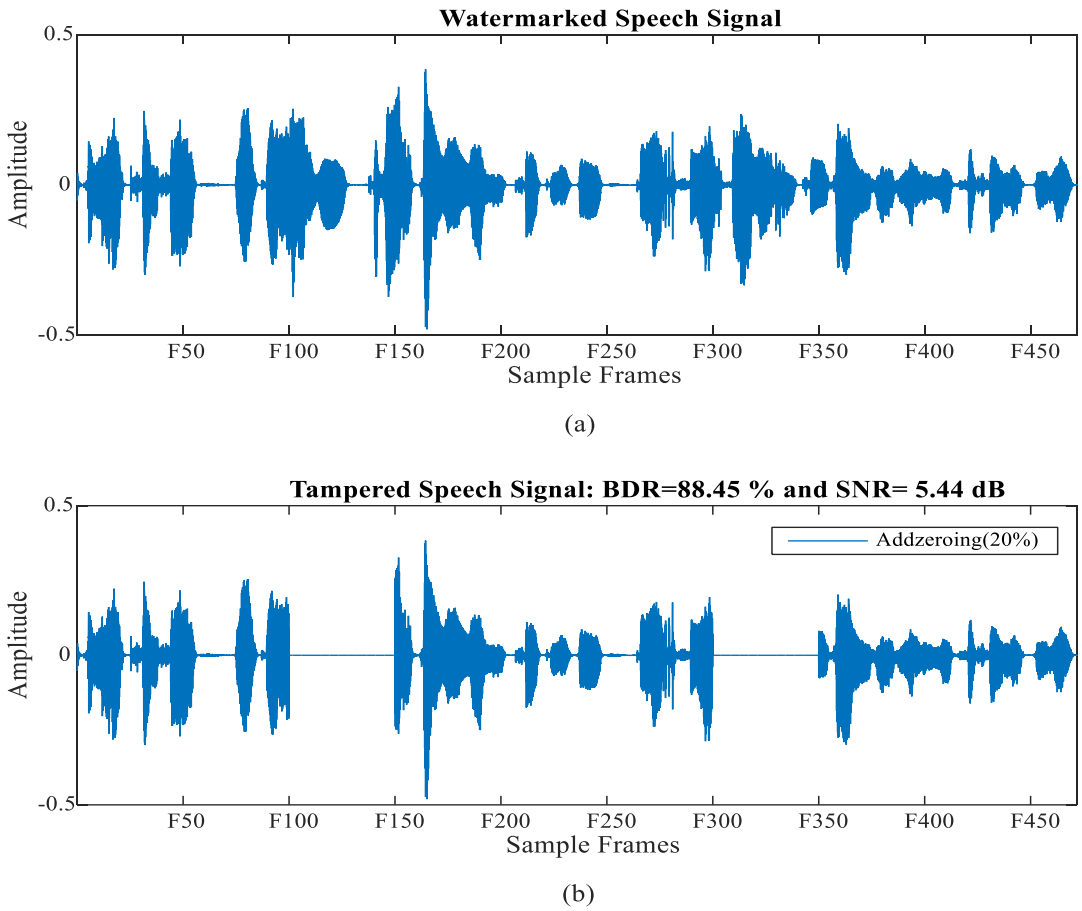


Figure 4.5: Waveform of the tampered speech by zeroing attack

(2) Adding noise

Noise is an undesirable thing that will be mostly encountered during the signal creation and transmission. In audio recordings and broadcast systems, audio noise refers to the residual low-level sound (four major types: hiss, rumble, crackle, and hum) that is heard in quiet periods of program. This variation from the expected pure sound or silence can be caused by the audio recording equipment, the instrument, or ambient noise in the recording room.

The presence of noise will surely degrade the quality of the signal. In this system, a white Gaussian noise (AWGN) is added to the test watermarked speech signals by keeping the SNRs of -40dB, -20dB, 20 dB, and 40 dB, respectively. AWGN is a random signal having a constant density and Gaussian amplitude distribution. If there is no signal on TV or radio, or sound of fun, its sound is similar to adding noise.

The resulting BDR values for different noise addition attacks are shown in Table 4.8. The lower SNR means the stronger noise and thus the lower speech quality. It can be evident in Figure 4.6 which shows a graphical illustration of the effects of different noise levels (-20% and 20% as an example) on a watermarked test speech signal. Figure 4.7 shows a comparative analysis of BDR results for different noise-adding attacks.

Table 4.8: Fragility of the proposed method against noise addition

Signal	BDR (%)			
	SNR=-40 dB	**SNR=-20 dB**	**SNR=20 dB**	**SNR=40 dB**
w_news1(f,B)	50.17	50.04	49.95	49.93
w_news9(f,B)	50.09	49.98	49.96	49.93
w_news19(m,B)	50.12	49.99	49.96	49.94
w_news22(f,E)	50.05	49.98	49.94	49.92
w_news38(m,E)	50.06	49.99	49.98	49.93
Average	**50.10**	**50.00**	**49.96**	**49.93**

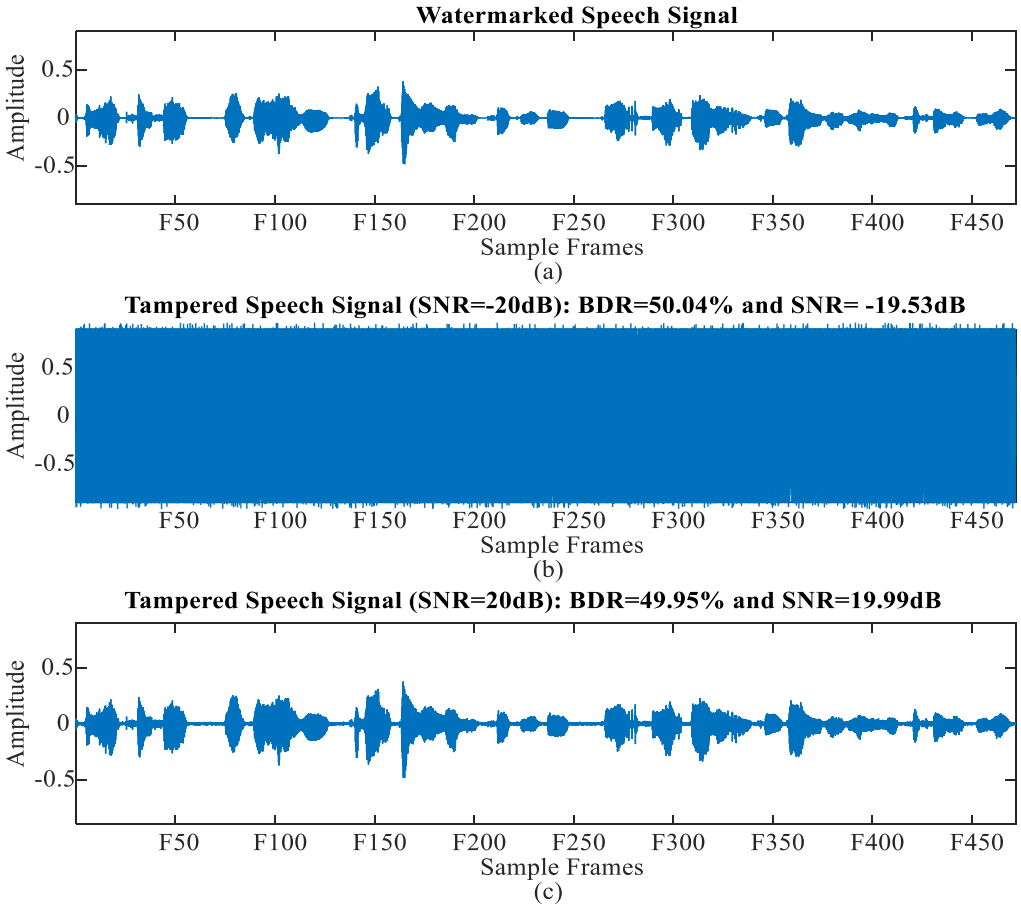


Figure 4.6: Waveforms of the tampered speeches by noise addition attack

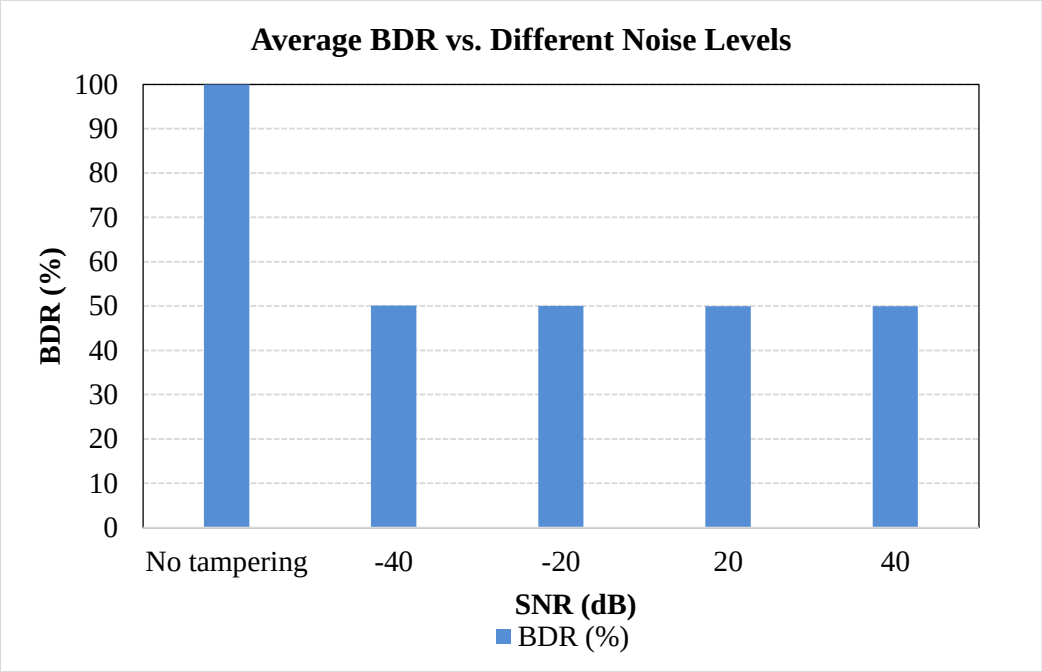


Figure 4.7: Comparative analysis of different noise addition attacks

It is found out that the waveform of the tampered speech with -20dB noise has completely different structure than the waveform of the original speech and it confirms a strong noise. For 20dB noise, the waveform is almost the same as the original and confirms a weak noise. Even for a weak noise, the average BDR is 49.95%, which is a strong indication of fragility. The effect of noise can also be evident by listening to the tampered speeches.

One interesting thing in Table 4.8 is that no matter how severe the noise is, there is no significant difference in BDR values. It is because when the noise is added to a signal, it is equally distributed throughout the signal. So, all noise levels, regardless of its strength, equally affect the signal.

(3) Reverberation

In audio/speech signal processing, reverberation (echo) is a reflection of sound that arrives at the listener with a delay after the direct sound. The delay is proportional to the distance of the reflecting surface from the source and the listener. Typical examples are the echo produced by the bottom of a well, by a building, or by the walls of an enclosed room and an empty room. A true echo is a single reflection of the sound source.

An echoed speech can be generated by first creating a delayed version of the original speech and then adding it back to the signal itself as mentioned in Eq. (4.4).

$$x(n) = s(n) + \alpha s(n - d), \tag{4.4}$$

where $\boldsymbol{s}(\boldsymbol{n})$ is the original speech, $\boldsymbol{s}(\boldsymbol{n}\text{-}\boldsymbol{d})$ is the delayed $\boldsymbol{s}(\boldsymbol{n})$, and $\boldsymbol{x}(\boldsymbol{n})$ is the echoed speech. The $\boldsymbol{d}= \boldsymbol{\tau F_s}$ is the delay in sampling interval given the delay $\boldsymbol{\tau}$ in seconds and sampling rate $\boldsymbol{F_s}$ in Hz, and $\boldsymbol{\alpha}$ is an attenuation factor.

To simulate reverberation attack in this system, different echoes generated by keeping $\boldsymbol{\alpha}$ =0.7 and $\boldsymbol{\tau}$=100 ms, 500 ms, 2000 ms, and 4000 ms, respectively, are added to the watermarked test speeches sampled at 8kHz. For example, for $\boldsymbol{\tau}$=100 ms, echo will be added after 800 samples ($\boldsymbol{d}$=100m×8k). For $\boldsymbol{\tau}$=4000 ms, echo will be added after 32000 samples ($\boldsymbol{d}$=4000m×8k). The longer the delay, the less the echo affect the signal.

Figure 4.8 shows the waveforms of the echo-affected speech signals (after 100ms and 4000ms). It can be seen from the waveforms that the structure of the echo-affected speech after 100ms is different from the structure of the original signal at the very first frames. As for the signal with echo effect after 4000ms, the structure of the waveform is starting to be different from the original one at F250 (250th frame). Even for 4000 ms delay which generally affects only 57% of a signal, the average BDR is 76.52%, which shows good fragility. Figure 4.9 is a comparative result of the different reverberation attacks.

Table 4.9: Fragility of the proposed method against reverberation

Signal		**Delay Time (τ)**			
		100 ms	**500 ms**	**2000 ms**	**4000 ms**
w_news1(f,B)	BDR (%)	50.67	53.36	63.29	76.52
	SNR (dB)	-0.36	-0.29	0.35	1.75
w_news9(f,B)	BDR (%)	50.43	53.10	62.28	74.44
	SNR (dB)	-0.32	-0.27	0.40	2.50
w_news19(m,B)	BDR (%)	50.46	52.65	60.31	70.87
	SNR (dB)	-0.42	-0.17	0.32	1.68
w_news22(f,E)	BDR (%)	50.46	52.65	60.31	70.87
	SNR (dB)	-0.41	-0.07	1.07	2.24
w_news38(m,E)	BDR (%)	50.44	52.13	58.51	68.15
	SNR (dB)	-0.38	-0.23	0.38	1.78
Average	**BDR (%)**	**50.49**	**52.78**	**60.49**	**72.01**
	SNR (dB)	**-0.38**	**-0.21**	**0.50**	**1.80**

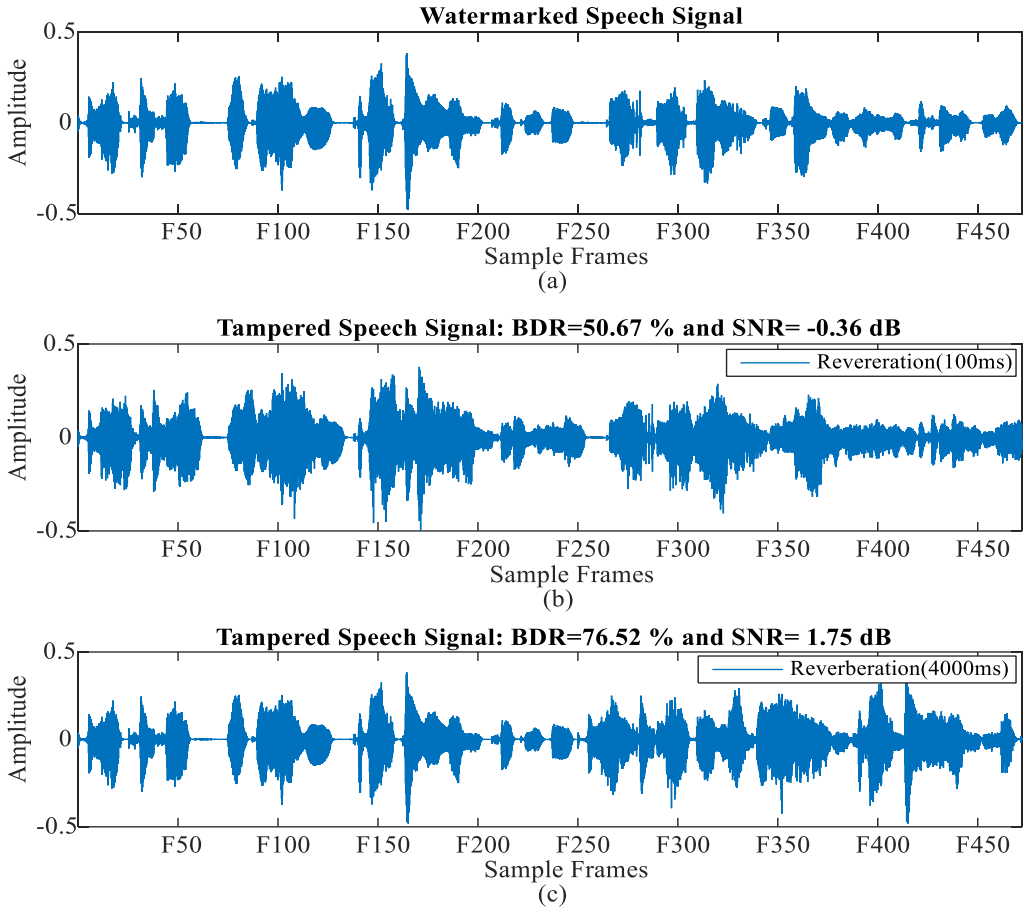


Figure 4.8: Waveforms of the tampered speeches by reverberation attack

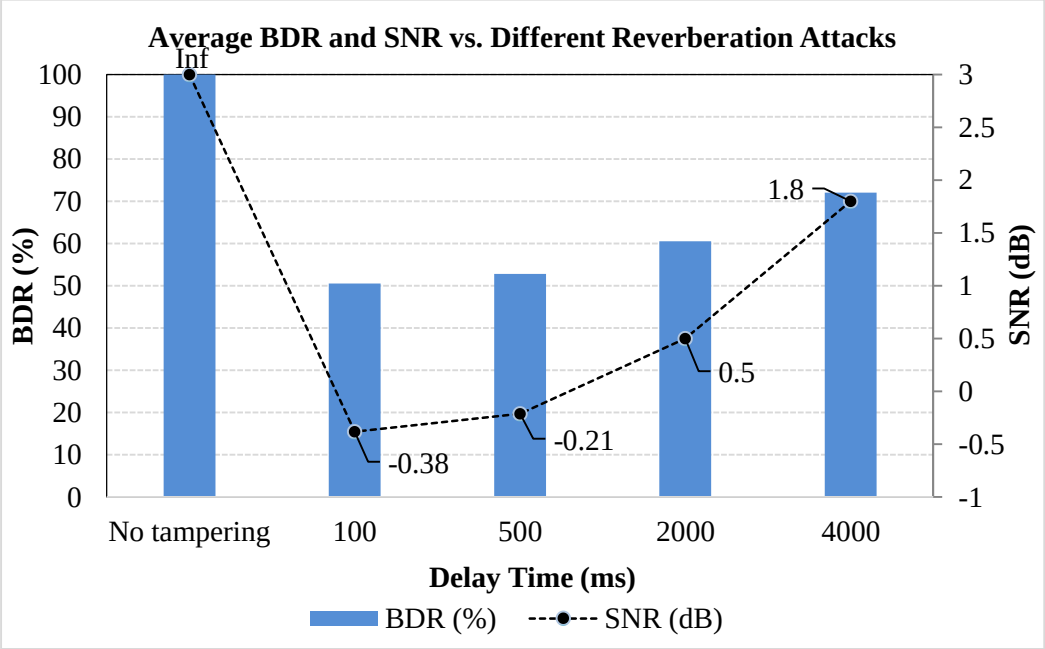


Figure 4.9: Comparative analysis of different reverberation attacks

(4) Concatenation

Concatenation is also one of the commonly found speech tampering types. For example, in digital forensics, the words in the speech evidence may be illegally concatenated (or may be replaced) with other words to divert the judgment.

To simulate concatenation attack in this system, the test watermarked speeches are segmented and some segments are replaced with un-watermarked speeches. It can be seen that the values of BDR and SNR are decreasing when the concatenation percentage is increasing, which is a sign of good fragility. Figure 4.10 is a comparative analysis of the effects of different concatenation attacks.

Table 4.10: Fragility of the proposed method against concatenation

Signal		**Concatenation Percentage**			
		20%	**40%**	**60%**	**80%**
w_news1(f,B)	BDR (%)	90.06	80.03	70.04	60.04
	SNR (dB)	55.53	52.51	50.75	49.50
w_news9(f,B)	BDR (%)	90.02	80.01	69.96	59.97
	SNR (dB)	57.50	54.50	52.73	51.50
w_news19(m,B)	BDR (%)	89.98	79.97	69.96	59.95
	SNR (dB)	56.80	53.77	51.99	50.73
w_news22(f,E)	BDR (%)	89.97	79.89	69.85	59.85
	SNR (dB)	57.11	54.04	52.26	51.03
w_news38(m,E)	BDR (%)	90.05	79.99	70.02	59.99
	SNR (dB)	51.49	48.47	46.70	45.44
Average	**BDR (%)**	**90.02**	**79.98**	**69.97**	**59.96**
	SNR (dB)	**55.69**	**52.66**	**50.89**	**49.64**

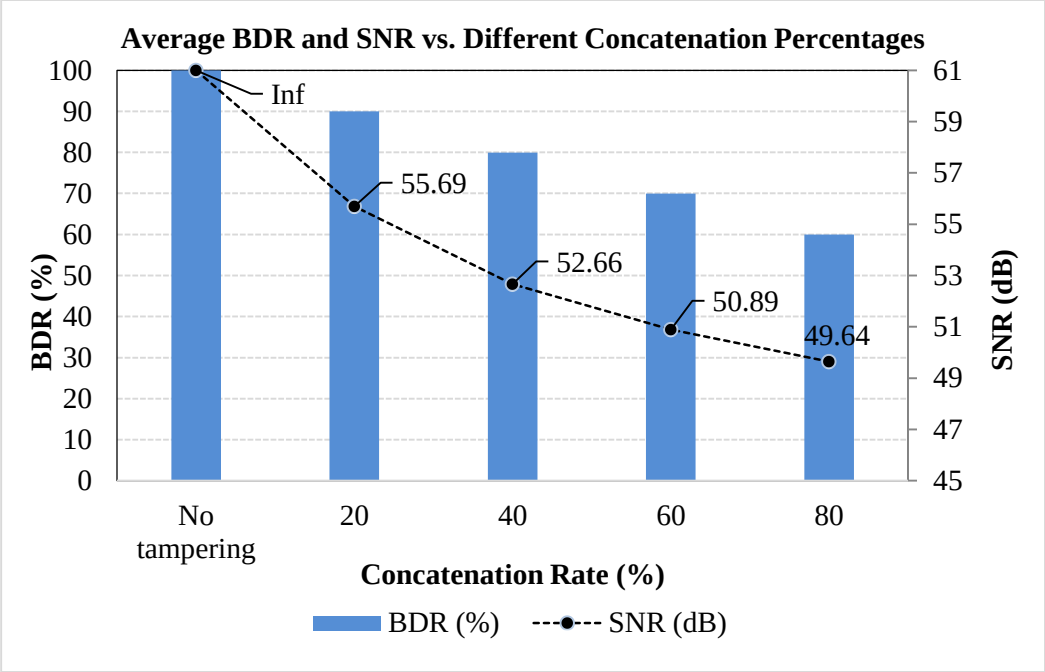


Figure 4.10: Comparative analysis of different concatenation attacks

(5) Time scaling

Time scaling on a signal $\boldsymbol{x}(\boldsymbol{t})$, e.g. $\boldsymbol{x}(2\boldsymbol{t})$ or $\boldsymbol{x}(\boldsymbol{t}/2)$, is related by linear scale changes in the independent variable, the time. If the signal $\boldsymbol{x}(\boldsymbol{t})$ is considered as a tape recording, then $\boldsymbol{x}(2\boldsymbol{t})$ is that recording played at twice the speed and $\boldsymbol{x}(\boldsymbol{t}/2)$ is the recording played at half-speed [6].

In this system, time scaling is simulated in MATLAB by using the built-in commands *upsample*(*x*,*n*) and *downsample*(*x*,*n*) [44]. If *x* is assumed as a speech file, the *upsample*(*x*,*n*) increases the sampling rate of the *x* by inserting *n*-1 zeros between samples, which yields the longer duration and thus slowing down the speech. The *downsample*(*x*,*n*) decreases the sampling rate of *x* by keeping every n^{th} sample starting with the first sample and discarding the others, which yields the shorter duration and thus speeding up the speech. Table 4.11 and Table 4.12 present the results of time scaling attacks with scale factor *n*=1.5, 2, 2.5, and 3, respectively. It can be seen that the larger the scale factor value, the lower the SNR.

Figure 4.11 and Figure 4.12 show the comparative analysis of the effects of different time scaling attacks (speed up and speed down). In addition, Figure 4.13 and 4.14 show the waveforms illustrating how the time scaling attacks look like.

Table 4.11: Fragility of the proposed method against time scaling (speed up)

Signal		Speed Up Factor			
		1.5	**2**	**2.5**	**3**
w_news1(f,B)	BDR (%)	49.06	48.82	48.62	48.45
	SNR (dB)	-20.78	-22.52	-23.30	-23.76
w_news9(f,B)	BDR (%)	49.21	48.71	48.72	48.44
	SNR (dB)	-18.83	-20.55	-21.33	-21.78
w_news19(m,B)	BDR (%)	49.32	48.82	48.52	48.48
	SNR (dB)	-19.59	-21.32	-22.10	-22.55
w_news22(f,E)	BDR (%)	49.35	48.95	48.63	48.57
	SNR (dB)	-19.15	-21.11	-21.96	-22.45
w_news38(m,E)	BDR (%)	49.49	48.90	48.76	48.93
	SNR (dB)	-17.45	-20.38	-21.49	-20.38
Average	**BDR (%)**	**49.29**	**48.84**	**48.65**	**48.57**
	SNR (dB)	**-19.16**	**-21.17**	**-22.03**	**-22.18**

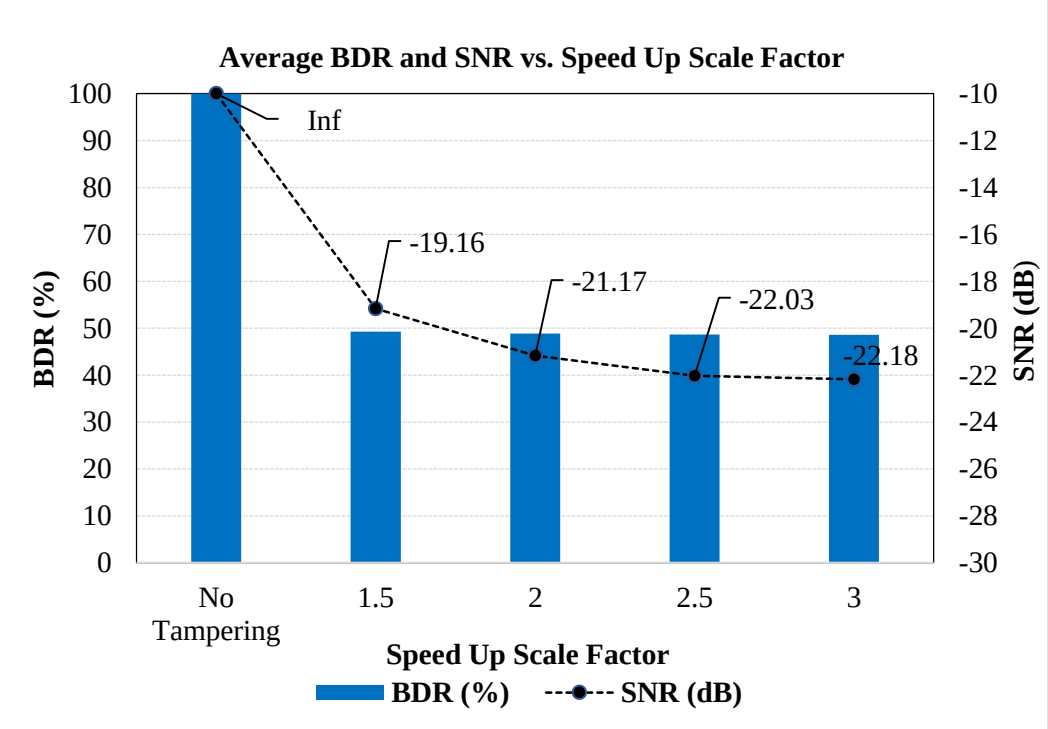


Figure 4.11: Comparative analysis of time scaling attack (speed up) with different scale factors

Table 4.12: Fragility of the proposed method against time scaling (speed down)

Signal		Speed Down Factor			
		1.5	**2**	**2.5**	**3**
w_news1(f,B)	BDR (%)	49.98	50.00	50.19	50.00
	SNR (dB)	-3.99	-4.77	-5.46	-6.02
w_news9(f,B)	BDR (%)	50.07	49.92	49.78	50.06
	SNR (dB)	-3.97	-4.74	-4.82	-6.09
w_news19(m,B)	BDR (%)	50.07	49.92	49.94	49.96
	SNR (dB)	-3.97	-4.74	-5.49	-6.44
w_news22(f,E)	BDR (%)	50.01	49.92	50.01	50.05
	SNR (dB)	-4.33	-5.16	-5.84	-6.44
w_news38(m,E)	BDR (%)	49.92	4.91	50.08	49.00
	SNR (dB)	-3.91	-4.69	-5.34	-5.94
Average	**BDR (%)**	50.01	40.92	49.96	49.82
	SNR (dB)	-4.03	-4.82	-5.39	-6.19

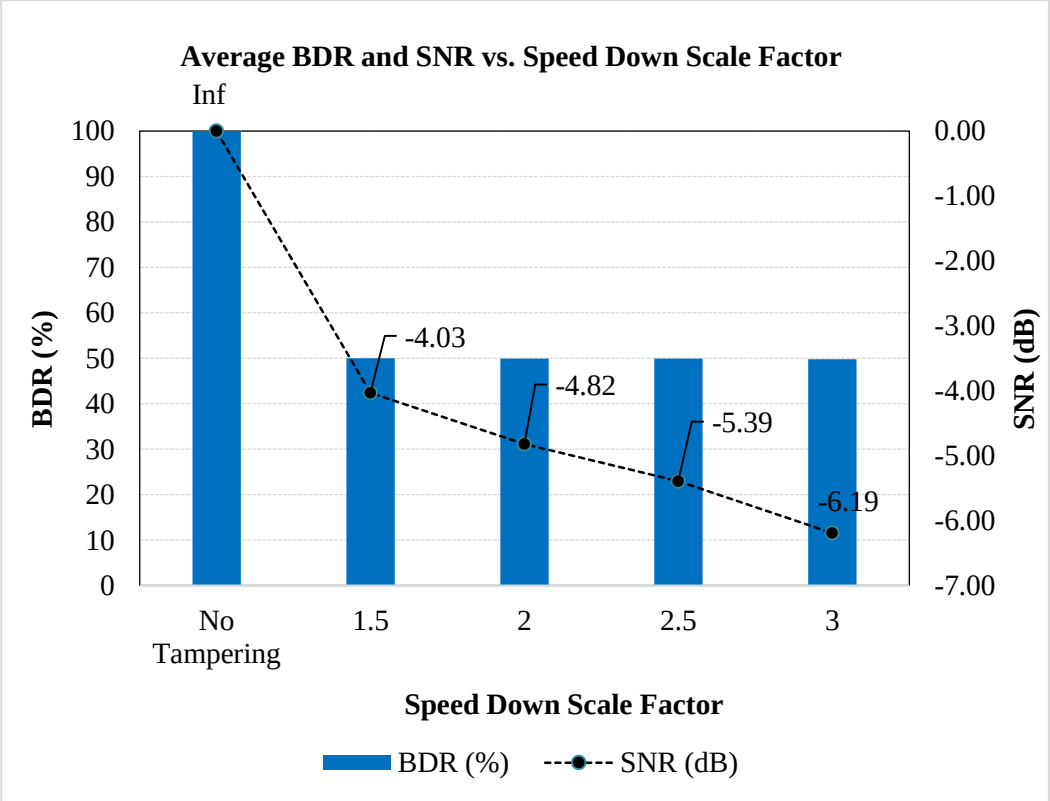


Figure 4.12: Comparative analysis of time scaling attack (speed down) with different scale factors

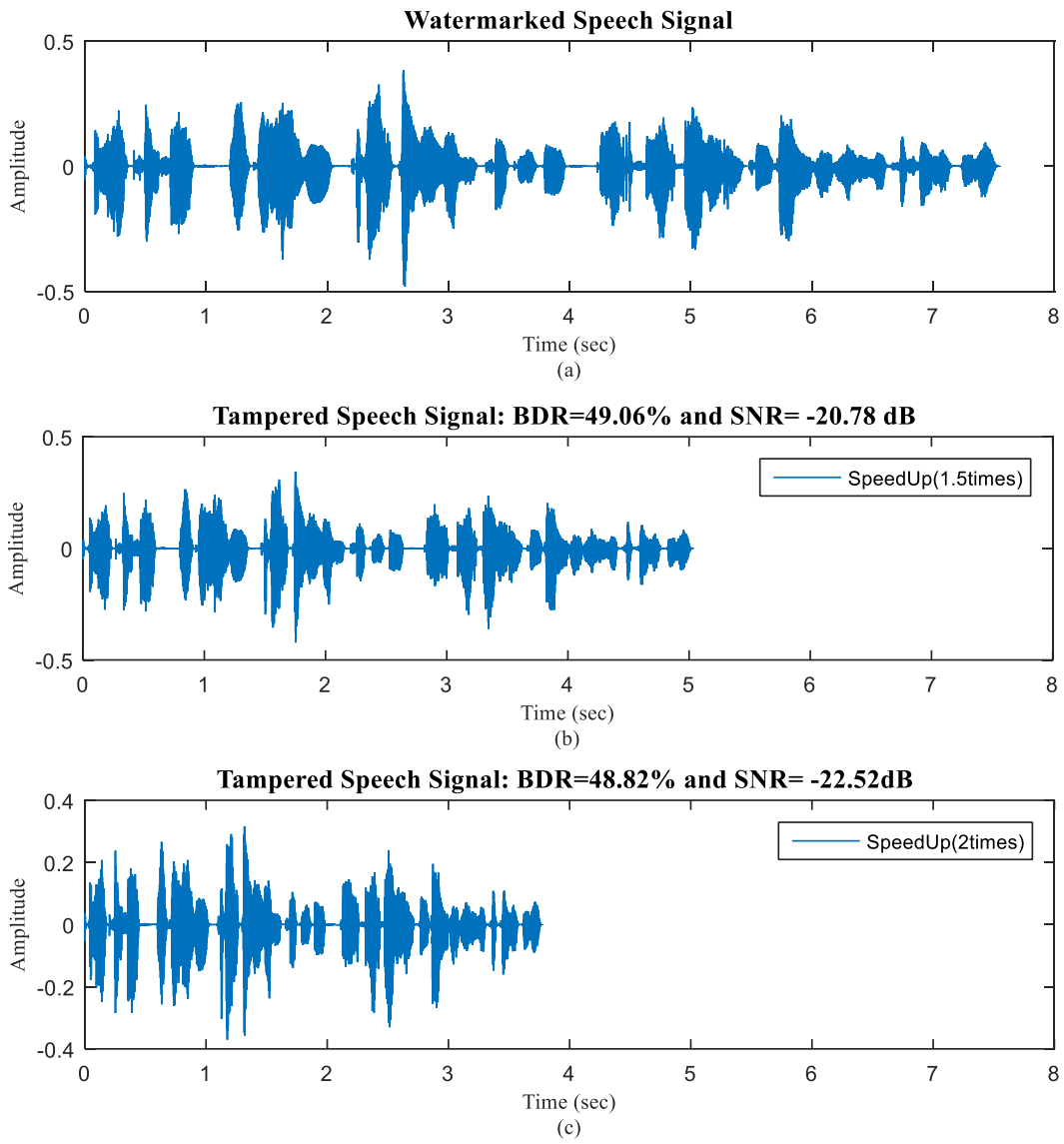


Figure 4.13: Waveforms of the tampered speeches by time scaling (speed up) attack

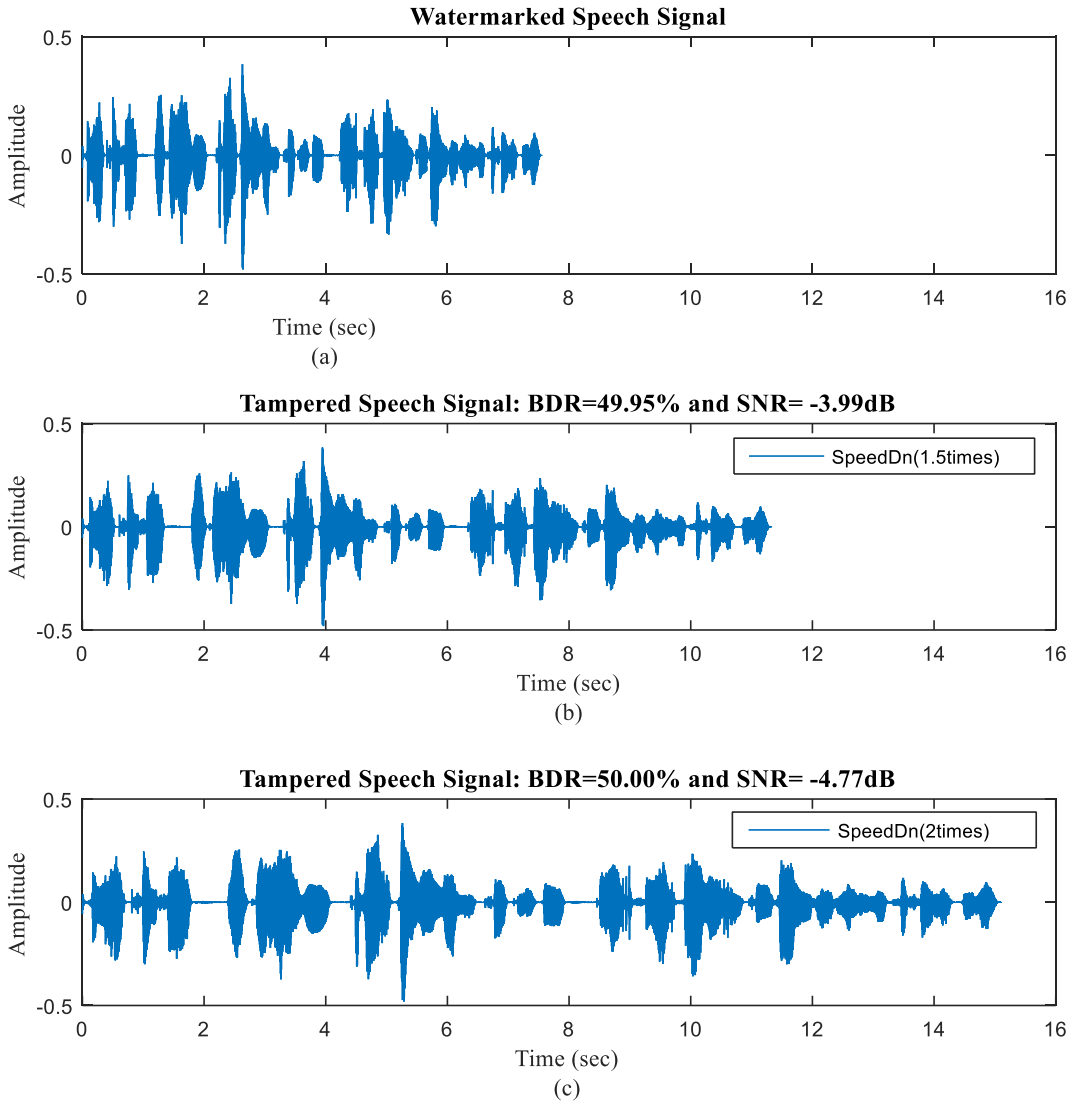


Figure 4.14: Waveforms of the tampered speeches by timescaling (speed down) attack

(7) Compression

Any multimedia data can be compressed in order to save memory/storage space, or to reduce in either time to transmit or in the amount of bandwidth required to transmit. Speech codec used to compress speech is a kind of necessary processing for speech transmission over the Internet and telecommunication systems. There are two types of compression: lossless and lossy [17], and lossy compression is commonly used for speech data. The reason why is that all parts of speech cannot be heard by human ear. Lossy compression techniques like MP3 and G.711 codecs take advantage of that fact by encoding sound parts that are less significant for perception using less bits. Those techniques are popular as they

significantly reduce the file size while maintaining good signal quality. In the field like digital forensics, illegal persons may intentionally try to compress the evident speech in order to remove some important parts.

In this system, one typical speech codec G.711, supported by MATLAB, is applied on the test watermarked speeches to evaluate the fragility of the proposed method [18]. G.711 is implemented by International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) recommendation for encoding, decoding, or converting speech signals. MATLAB G.711 codec block is a logarithmic scalar quantizer designed for narrowband speech, which is defined as a voice signal with an analog bandwidth of 4 kHz and a Nyquist sampling frequency of 8 kHz. The block quantizes a narrowband speech input signal by using A-law or mu-law so that it can be transmitted using only 8-bits (original speech is 16-bit encoded).

In this experiment, the watermarked speeches are compressed at rate from 128 kbps to 64 kbps with G.711. The SNR and BDR results for G.711 compression are shown in Table 4.13 and the corresponding graphical illustration are shown in Figure 4.15 and Figure 4.16, for A-law and mu-law respectively. For all signals, the BDR results are around 50% after compressing the files to half of their original size, which satisfy the criteria BDR≤ 90% and thus a good sign of fragility. Figure 4.17 shows how the compression affects a signal.

Table 4.13: Fragility of the proposed method against G.711 compression

Signal		**Compression (G.711)**	
		A-law	**μ-law**
w_news1(f,B)	BDR (%)	56.60	49.90
	SNR (dB)	-19.92	-20.20
w_news9(f,B)	BDR (%)	56.59	49.95
	SNR (dB)	-17.80	-18.05
w_news19(m,B)	BDR (%)	55.79	50.16
	SNR (dB)	-19.21	-19.00
w_news22(f,E)	BDR (%)	50.22	50.02
	SNR (dB)	-18.77	-18.19
w_news38(m,E)	BDR (%)	50.38	49.76
	SNR (dB)	-24.80	-23.80
Average	**BDR (%)**	**53.92**	**49.96**
	SNR (dB)	**-20.10**	**-19.85**

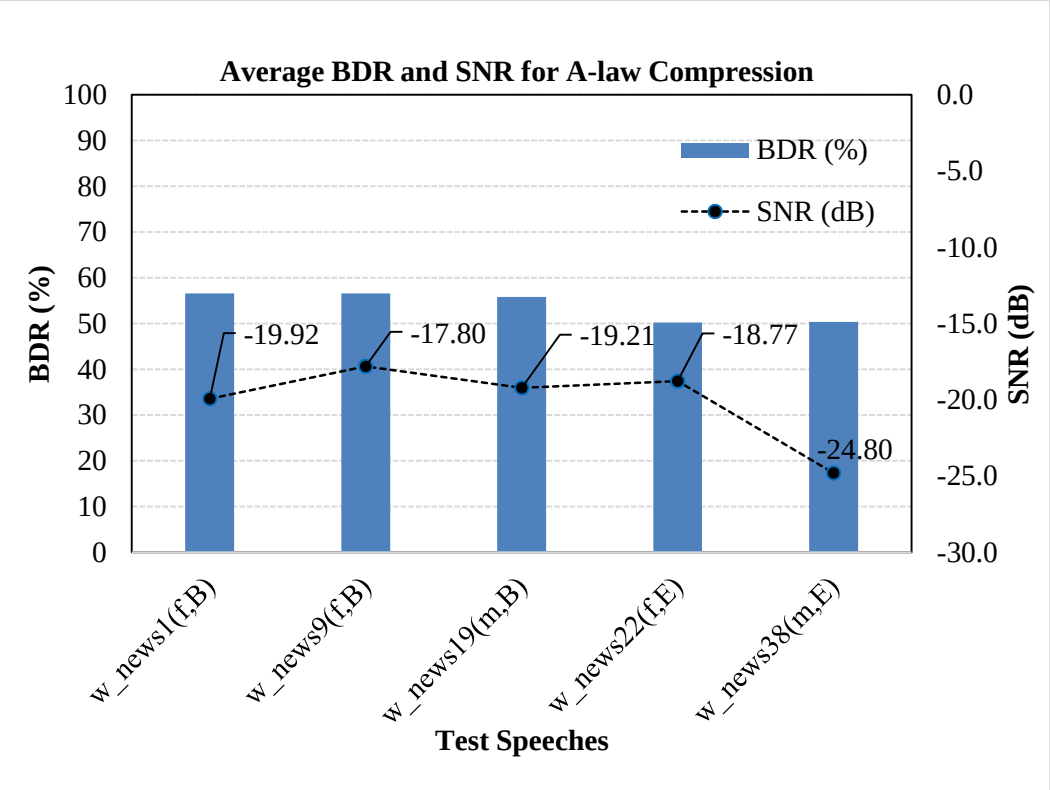


Figure 4.15: Comparative analysis of compression attack with G.711 (A-law)

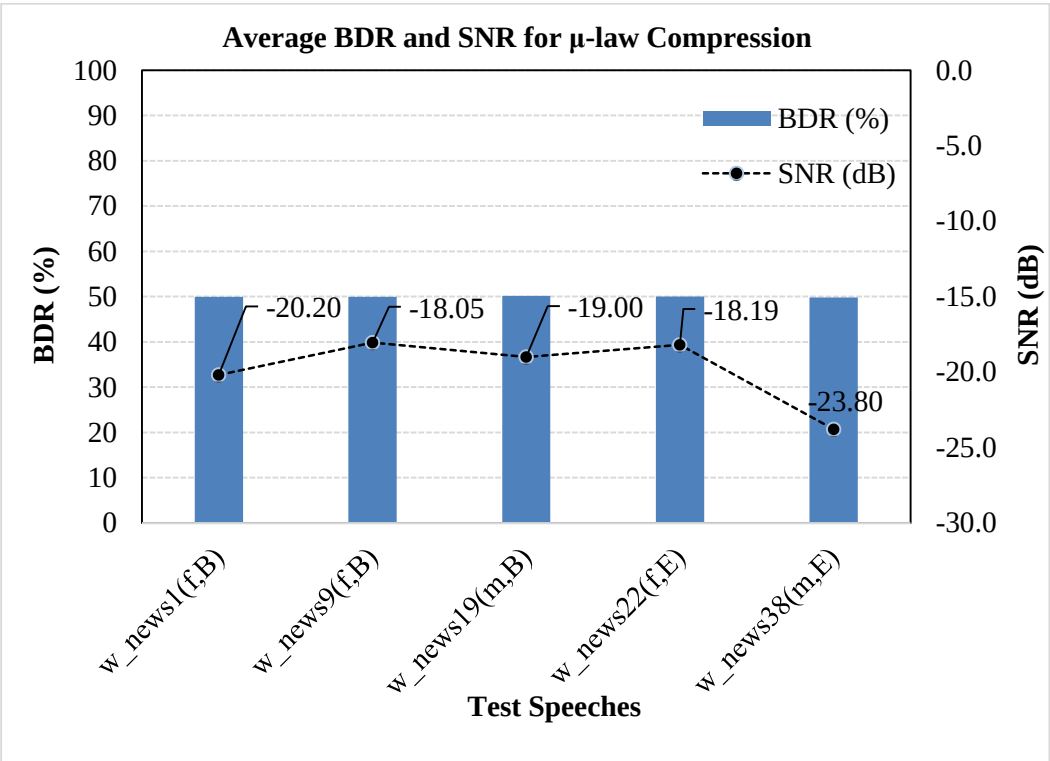


Figure 4.16: Comparative analysis of compression attack with G.711 (μ-law)

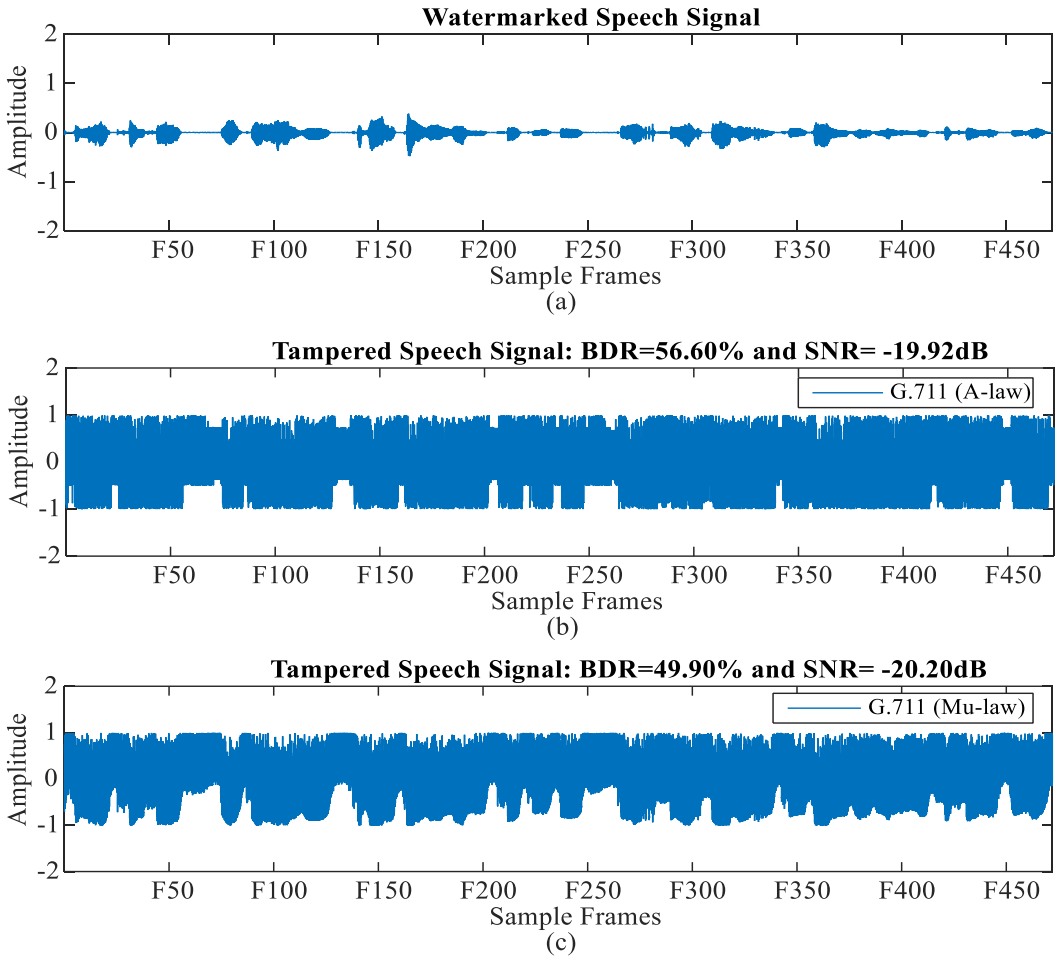


Figure 4.17: Waveforms of the tampered speeches by G.711 compression

4.2 Tampering Localization

Tampering localization is very important for applications which need not only to check the integrity of the received speech but also the entire healthy speech. By localizing the tampering regions, it can avoid the retransmission of the whole signal.

In this system, for clear illustration, tampering regions are localized by denoting one and zero respectively for tampered and not-tampered frames on a graph. Figure 4.18 (a) shows the waveform of a 7-sec long un-watermarked speech signal. Figure 4.18 (b) shows the waveform of the watermarked speech. It can be observed that the waveform of the watermarked speech looks similar to the waveform of its respective original speech and differences are not perceivable. Therefore, they do not attract the attention of attackers. Figure 4.18 (c) shows the waveform of the tampered speech by a malicious attacker. As an

example, zeroing attack is applied in which 21% of the watermarked speech is replaced by silence (zero). Since the proposed scheme is a frame-based watermarking, it is noticeable that watermarks in the tampered frames were destroyed. Figure 4.18 (d) shows the results of the hash bit examination procedure in determining the tampered frames, in which 0's shows the reserved frames and 1's shows the tampered identified frames. In this way, the tampered regions can be easily localized. Figure 4.19 and Figure 4.20 show the tampering localization maps for reverberation attack after 4000ms and the noise addition attack by keeping SNR of 20dB, respectively.

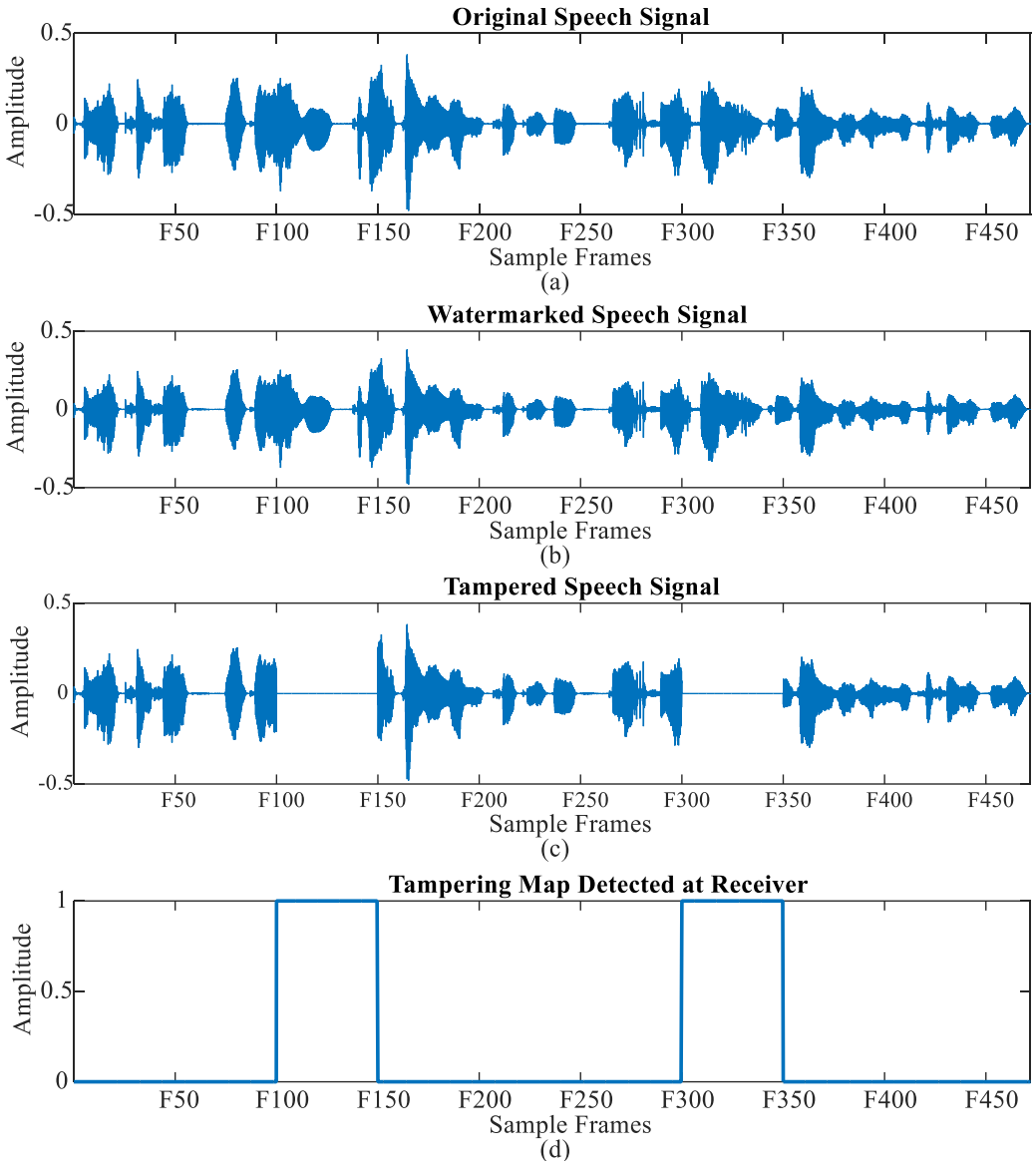


Figure 4.18: Tampering localization map for zeroing attack

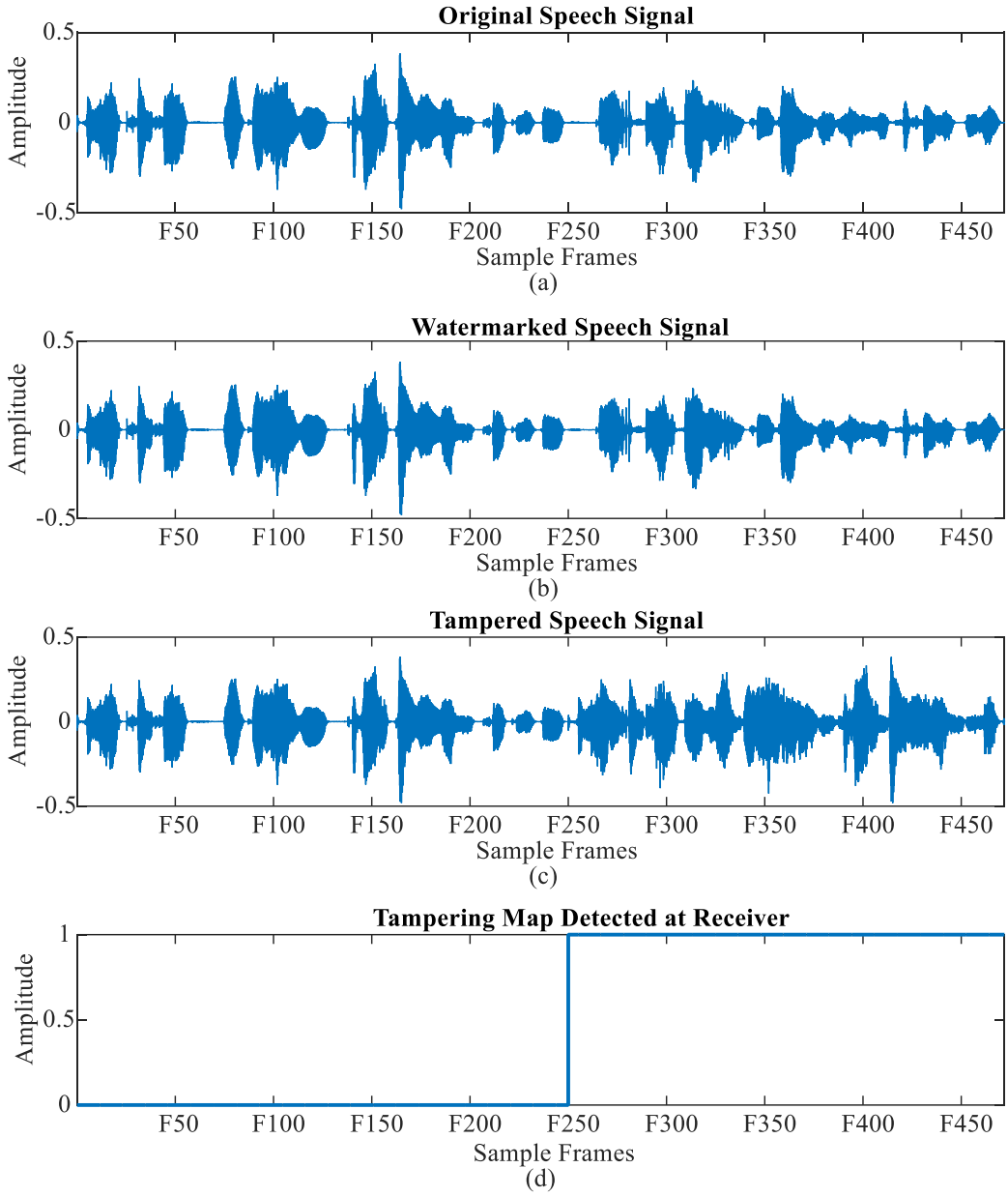


Figure 4.19: Tampering localization map for reverberation attack

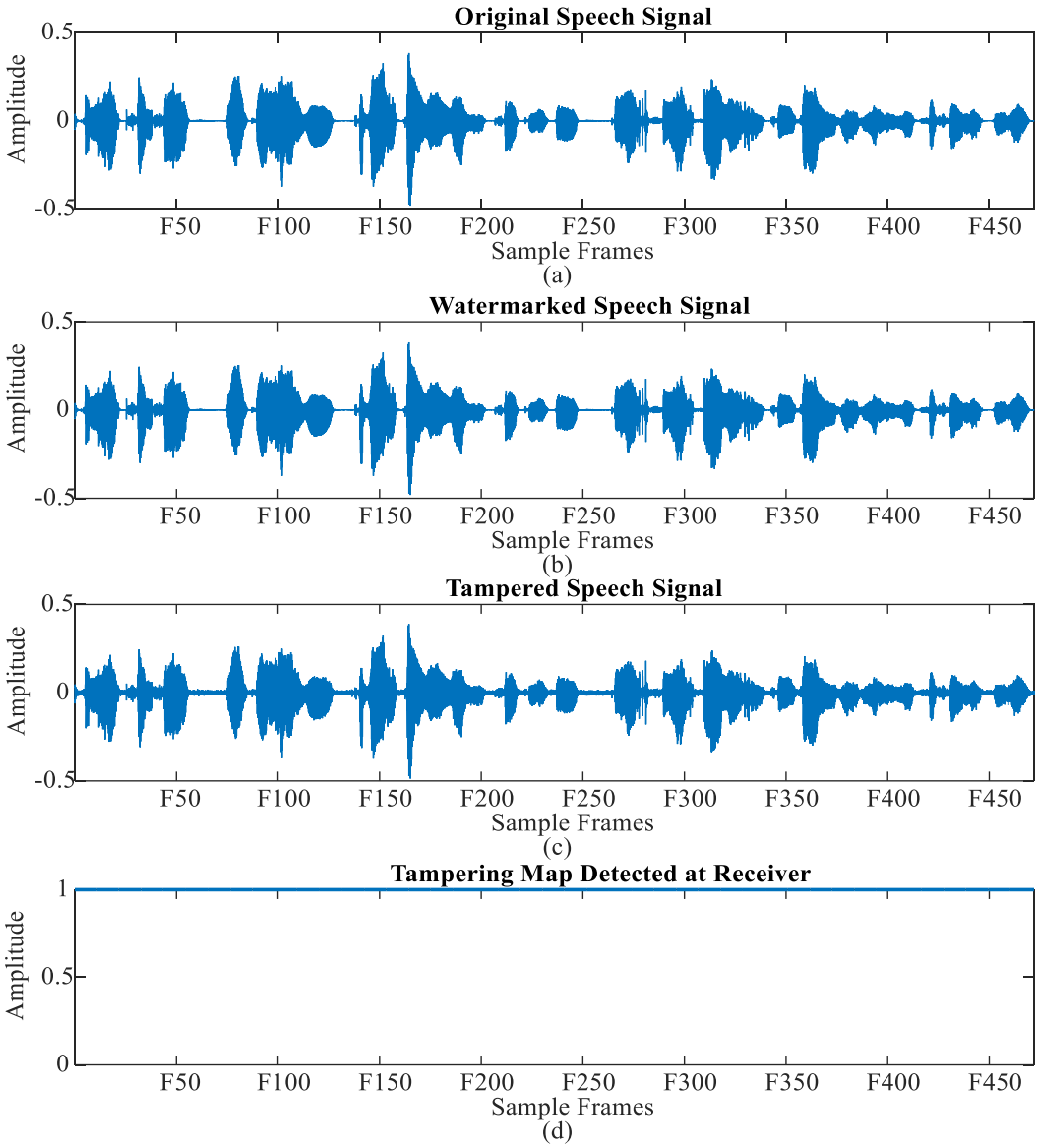


Figure 4.20: Tampering localization map for noise addition attack

4.3 Summary

This chapter discussed the evaluation results of the performance of the proposed method with respect to inaudibility and fragility. Firstly, it discussed how many LSB bits should be used for watermark embedding while keeping good inaudibility. Then, various kinds of most commonly found tampering types such as zeroing, noise addition, etc. are applied on the test watermarked speeches and evaluated the fragility of the proposed method. According to the experimental results, the proposed speech watermarking method achieves both satisfying inaudibility and fragility test results.

CHAPTER 5

CONCLUSION

With the advance of versatile digital multimedia processing tools, speech signals can be easily duplicated and manipulated by unauthorized users. Since the speech is an important information carrier not only in our daily life but also for more important areas such as governmental and commercial activities, integrity and authenticity of speech is very important.

In this thesis, an efficient tampering detection and localization method for speech signals is proposed by utilizing the SHA-512 hash algorithm and the LSB replacement watermarking method. A self-embedding speech signal is produced by inserting a watermark that consists of a representation of the original signal into itself to show fragility against tampering. The hash information of the signal frames acted as the watermark and helped the receiver to distinguish between the healthy (reserved) and tampered (erased) frames. This system is intended to reduce the cost and time of the retransmission of the entire speech signal by locating tampering regions, if the signal is tampered.

The SNR, LSD, and BDR measures were used to evaluate the fragility and inaudibility performance of the proposed system. According to the SNR and LSD results shown in Table 4.2 to 4.6 of Chapter 4, the proposed system achieved high data embedding capacity and acceptable inaudibility results. The BDR results for different tampering types of noise addition, zeroing, reverberation, concatenation, time scaling, and compression showed that the proposed system was fragile enough to detect and locate the tampering regions precisely.

5.1 Further Extension

For future work, the proposed method can be extended to support the self-recovery feature that can recover the tampered speech with proper speech quality for high tampering rates, i.e. as if it has the same quality as the original speech signal. In the self-recovery schemes, a watermark generated from the original signal content is embedded into itself to combat the tampering situations, in which a part of the original signal is modified maliciously. The amount of the watermark that survives the tampering helps the receiver not only to detect

the tampering and localize it, but also to recover the lost content with a certain quality, depending on the tampering rate and the structure applied for the watermark generation. By providing self-recovery feature, it can reduce the time and cost needed for retransmission of the signal.

In addition, future research can be carried on for watermark embedding in video sequences i.e. movies or surveillance. Applying watermarking technique on a surveillance system will decrease the security issues by keeping track of the voice communication.

REFERENCES

[1] A. P. F. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding-a survey,” Proc. IEEE, vol. 87, no. 7, pp. 1062-1078, 1999.

[2] “An overview of cryptography,” Date of access: April 2016. http://www.garykessler.net/library/crypto.html/hash.

[3] B. Lei, I. Y. Soon, F. Zhou, Z Li, and H. Lei, “A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition,” Signal Processing, vol. 92, no. 9, pp. 1985-2001, 2012.

[4] B. Li, J. He, J. Huang, and Y. Shi, “A survey on image steganography and steganalysis,” Journal of Information Hiding and Multimedia Signal Processing (JIHMSP), vol. 2, no. 2, pp. 142-172, April 2011.

[5] B. S. Ko, R. Nishimura, and Y. Suzuki, “Time-spread echo method for digital audio watermarking,” IEEE Trans. Multimedia, vol. 7, no. 2, pp. 212-221, 2005.

[6] “Basic signal operations in DSP: time shifting, time scaling, and time reversal,” Date of access: May 2017. https://www.allaboutcircuits.com/technical-articles/basic-signal-operations-in-dsp-time-shifting-time-scaling-and-time-reversal.

[7] C. D. Roover, C. D. Vleeschouwer, F. Lefebvre, and B. Macq, “Robust video hashing based on radial projections of key frames,” IEEE Trans. Signal Processing, Supplement on Secure Media vol. 53, pp. 4020-4037, 2005.

[8] C. Fei, D. Kundur, and R. H. Kwong, “Analysis and design of secure watermark-based authentication systems,” IEEE Trans. Information Forensics and Security, vol. 1, no. 1, pp. 43-55, 2006.

[9] C. I. Podilchuk and E. J. Delp, “Digital watermarking: algorithms and applications,” IEEE Signal Processing Magazine, vol. 18, no. 4, pp. 33-46, 2002.

[10] C. P. Wu and C. C. J. Kuo, “Fragile speech watermarking based on exponential scale quantization for tamper detection,” Proc. ICASSP, vol. IV, pp. 3305-3308, 2002.

[11] C. R. Abbey and H. H. Pursel, “Data channel monitor,” United States Patent, pp. 3,415,947, 1968.

[12] D. E. H. and C. M. Solar, “Automatic monitor for programs broadcast,” United States Patent, pp. 4,025,851, 1977.

[13] E. T. Lin, A. M. Eskicioglu, R. L. Lagendijk, and E. J. Delp, “Advances in digital video content protection,” Proc. IEEE, vol. 93, no. 1, pp. 171-183, 2005.

[14] F. Cayre, C. Fontaine, and T. Furon, “Watermarking security: theory and practice,” IEEE Trans. Signal Processing, Supplement on Secure Media, vol.53, pp. 3976-3987, 2005.

[15] F. Hartung and M. Kutter, “Multimedia watermarking techniques,” Proc. IEEE, vol. 87, pp. 1079-1107, 1999.

[16] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, “Watermarking digital image and video data: a state-of-the-art overview,” IEEE Signal Processing Magazine, vol. 17, no. 5, pp. 20-46, 2000.

[17] “G.711,” Date of access: September 2017. https://en.wikipedia.org/wiki/G.711.

[18] “G711 Codec,” Date of access: September 2017. https://www.mathworks.com/help/dsp/ref/g711codec.html.

[19] H. Bi, Y. Liu, Y. Ge, Y. Zhang, and M. Wu, “Watermark detection in NSCT-domain,” School of Electrical Information Engineering, Northeast Petroleum University, Daqing 163318, Chin.

[20] H. Hering, H. Martin, and G. Kubin, “Safety and security increase for air traffic management through unnoticeable watermark aircraft identification tag transmitted with the VHF voice communication,” in Proc. 22nd Digital Avionics Syst. Conf. (DASC ’03), vol. 1, pp. 4.E.2–41-10, October 2003.

[21] H. J. Kim and Y. H. Choi, “A novel echo-hiding scheme with backward and forward kernels,” IEEE Trans. Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 885-889, 2003.

[22] H. Kawahara, H. Banno, T. Irino, and P. Zolfaghari, “ALGORITHM AMALGAM: Morphing waveform based methods, sinusoidal models and STRAIGHT,” Proc. ICASSP, pp. 13-16, 2004.

[23] H. S. Malvar and A. F. Florncio, “Improved spread spectrum: A new modulation technique for robust watermarking,’ IEEE Trans. Signal Processing, vol. 51, no. 4, pp. 898-905, 2003.

[24] H. Yi and C. L. Philipos, “Evaluation of objective measures for speech enhancement,” Interspeech2006, pp. 1447-1450, September 2006.

[25] I. J. Cox, G. Dorr, and T. Furon, “Watermarking is not cryptography,” Lecture Notes in Computer Science, vol. 4283, pp. 1-15, Springer, 2006.

[26] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” IEEE Trans. Image Processing, vo1. 6, no. 12, pp.1673-1687, 1997.

[27] I. W. Evett, “Towards a uniform framework for reporting opinions in forensic science case-work,” Science & Justice, vol. 38, pp. 198-202, 1998.

[28] I. W. Evett, G. Jackson, J. A. Lambert, and S. McCrossan, “The impact of the principles of evidence interpretation on the structure and content of statements,” Science & Justice, vol. 40, pp. 233-239, 2000.

[29] J. G. Rodrguez, A. Drygajlo, D. R. Castro, M. G. Gomar, and J. O. Garca, “Robust estimation, interpretation and assessment of likelihood ratios in forensic speaker recognition,” Computer Speech and Language, pp. 331-355, 2006.

[30] J. G. Rodrguez, A. Drygajlo, D. R. Castro, M. G. Gomar, and J. O. Garca, “Robust estimation, interpretation and assessment of likelihood ratios in forensic speaker recognition,” Computer Speech and Language, pp. 331-355, 2006.

[31] M. A. Nematollahi and S. A. R. Al-Haddad, “An overview of digital speech watermarking,” International Journal of Speech Technology, Springer, 2013.

[32] M. Celik, G. Sharma, and A. M. Tekalp, “Pitch and duration modification for speech watermarking,” Proc. ICASSP, vol. II, pp. 17-20, 2005.

[33] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, “Multimedia data-embedding and watermarking technologies,” Proc. IEEE, vol. 86, no. 6, pp. 1064-1087, 1998.

[34] M. Fallahpour and D. Megias, “High capacity logarithmic audio watermarking based on the human auditory system,” IEEE International Symposium on Multimedia (ISM), pp. 28-31, 2012.

[35] M. Unoki and D. Hamada, “Method of digital-audio watermarking based on cochlear delay characteristics”, International Journal of Innovative Computing, Information and Control, vol. 6, no. (3(B)), pp. 1325-1346, 2010.

[36] M. Unoki and R. Miyauchi, “Reversible watermarking for digital audio based on cochlear delay characteristics,” Proc. IIHMSP2011, pp. 314–317, 2011.

[37] P. Bassia and I. Pitas, “Robust audio watermarking in the time domain,” Proc. EUSIPCO, pp. 25-28, 1998.

[38] R. Chowdhury, D. Bhattacharyya, S. K. Bandyopadhyay and T. Kim, “A view on LSB based audio steganography,” International Journal of Security and Its Applications, vol. 10, no. 2, pp. 51-62, 2016.

[39] R. G. Cole and J. H. Rosenbluth, “Voice over IP performance monitoring,” ACM SIGCOMM Computer Communication Review, vol. 31, no. 2, pp. 9-24, 2001.

[40] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, “A digital watermark,” Proc. IEEE International Conference on Image, vol. II, pp. 86-90, 1994.

[41] R. L. Rivest, “The MD4 Message Digest Algorithm, request for comments (RFC) 1320,” Internet Activities Board, Internet Privacy Task Force, April 1992.

[42] R. L. Rivest, “The MD5 Message Digest Algorithm, request for comments (RFC) 1321,” Internet Activities Board, Internet PrivacZ Task Force, April 1992.

[43] R. Poisel and S. Tjoa, “Forensics investigations of multimedia data: A Review of the State-of-the-Art,” Proc. IT Security Incident Management and IT Forensics (IMF), pp. 48-61, 2011.

[44] “Resample,” Date of access: October 2017. https://www.mathworks.com/signal/resampling.html.

[45] S. Khurana, “Watermarking and information-hiding,” International Journal of Computer Science and Information Technologies (IJCSIT), vol. 2, no. 4, pp. 1679-1681, 2011.

[46] S. Lingfen and E.C. Ifeachor, “Voice quality prediction models and their application in VoIP networks,” IEEE Trans. Multimedia, vol. 8, no. 4, pp. 809-820, 2006.

[47] S. Milani , M. Fontani , P. Bestagini , M. Barni , A. Piva , M. Tagliasacchi, and S. Tubaro, “An overview on video forensics,” APSIPA Trans. Signal and Information Processing, vol. 1, pp. 1-18, 2012.

[48] S. Sarreshtedari, M. A. Akhaee, and A. Abbasfar, “A watermarking method for digital speech self-recovery,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 23, pp. 1917-1925, Nov. 2015.

[49] S. Wang, N. S. Kim, and M. Unoki, “Formant enhancement based speech watermarking for tampering detection,” School of Information Science, JAIST, vol. 6, Sep. 2014.

[50] S. Wu, J. Huang, D. Huang, and Y. Q. Shi, “Efficiently self-synchronized audio watermarking for assured audio data transmission,” IEEE Trans. broadcasting, vol. 51, no. 1, pp. 69-76, 2005.

[51] “Secure Hash Algorithms,” Date of access: September 2016. https://en.wikipedia.org/wiki/Secure_Hash_Algorithms/SHA.

[52] “Secure Hash Algorithms,” Date of access: September 2016. https://en.wikipedia.org/wiki/Secure_Hash_Algorithms/SHA-1.

[53] “Secure Hash Algorithms,” Date of access: September 2016. https://en.wikipedia.org/wiki/Secure_Hash_Algorithms/SHA-2.

[54] T. Ohsawa and M. Karita “Automatic telecasting or radio broadcasting monitoring system,” United States Patent, pp. 3,760,275, 1973.

[55] T. Painter and A. Spanias, “Perceptual coding of digital audio,” Proc. IEEE, vol. 88, no. 4, pp. 451-515, 2000.

[56] T. Toda, A. W. Black, and K. Tokuda “Voice conversion based on maximum likelihood estimation of spectral parameter trajectory,” IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 8, pp. 2222-2235, 2007.

[57] “Voice over IP,” Date of access: November 2016. http://en.wikipedia.org/wiki/Voice-over-IP.

[58] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding,” IBM Systems Journal, vol. 35, no. 3/4, pp. 313-336, 1996.

[59] Y. Erfani and S. Siahpoush, “Robust audio watermarking using improved TS echo hiding,” Digital Signal Processing, vol. 19, no. 5, pp. 809-814, 2009.

PUBLICATION

[1] Sharr Wint Yee Myint and Twe Ta Oo, “An efficient tampering detection and localization method for speech signals,” Parallel and Soft Computing Conference, University of Computer Studies, Yangon, Myanmar, 2018.