Digital Watermarking-Based Authentication Techniques For Real-Time Multimedia Communication

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades Dr.-Ing.

vorgelegt von Song Yuan
aus Liaoning, China

Referenten: Prof. Dr.-Ing. Sorin A. Huss
Korreferent: Prof. Dr.-Ing. Ralf Steinmetz

Tag der Einreichung: 22. Juli 2005

Tag der Prüfung: 09. Dezember 2005

Darmstädter Dissertation D17

Abstract

Data integrity and source origin authentication are essential topics for real-time multimedia systems. But the traditional methods are not very applicable to overcome the distortion introduced in multimedia data transportation. In this thesis some security mechanics are proposed, which rely on authentication rather than on encryption methods. The highly asymmetric architectures found in ubiquitous computing applications are exploited to provide a protection of the transmitted multimedia data by means of well-known digital watermarking techniques. The advocated approach adopts public key encryption to efficiently generate non-repudiate multimedia data, whereas both the digital watermark generation and the authentication algorithms are public.

Acknowledgements

Great thanks to Prof. Sorin A. Huss, Prof. Ralf Steinmetz and Prof. Karsten Weihe!

Many thanks to my colleagues at Integrated Circuits and Systems Lab of TU Darmstadt!

Contents

Chapter 1 Introduction

1.1 Driving Force of Digital Watermarking Research

Digital cameras and video recorders are widely used nowadays. The digital pictures and video appear in the internet or are stored in portal storages such as DVDs and magnetic disks. The popularity of digital multimedia causes two serious problems: 1) the multimedia products are easily to download and reproduce for the commercial profit. That is the so-called intelligent property privacy; 2) with the fast adoption of the powerful multimedia manipulation tools, multimedia data such as pictures, video or audio clips have been decreased the credibility.

Authenticity has broad meanings [1.4]: 1) not false or imitation; 2) confirming to or based on facts; 3) no one has tampered with the contents; 4) something is actually from the original source. For multimedia protection, the third meaning is about integrity of multimedia data and people use the word “copyright protection" to indicate the fourth meaning.

So-called multimedia authentications are techniques for multimedia integrity verification and source origin authentication. The techniques are implemented by means of digital signature or digital watermarking. Digital signature is a non-repudiatible, encrypted message digest extracted from the content. It is usually stored as a separate file, which can be attached to the data to prove integrity and originality. On the contrary, digital watermarking takes the approach of inserting watermark into the multimedia data thus that the watermark is residing in the protected multimedia data.

Traditional digital signature is so vulnerable that a bit alteration in multimedia data or signature may disable the authentication. On the contrary, watermarking techniques are more flexible to survive some manipulations. Therefore, people look digital watermarking as a promising solution for multimedia protection.

Two recent published books written by I. Cox *et al.* [1.1] and M. Arnold *et al.* [1.2] give a complete literature survey of the state-of-the-art digital watermarking techniques. The main driving forces that led researchers to do work in this area are protection on the copyrights of digital multimedia, such as audio, images, and video. Due to the enormous increase of Internet data communications, the producers of such multimedia data feel “threatened” by the presence of attackers (also called hackers or pirates) who could get hold of copyrighted data and possibly sell them as their own products on the Internet or even

give them away for free (e.g., Napster). It has been reported that the audio industry, for example, loses billions of U.S. dollars every year due to pirated audio clips.

It is worth to be mentioned that, at the time of this writing, most ideas in this field are presented and the research of watermarking has gone into a bottleneck unless there is a breakthrough on human vision/audio system. Watermarking was once considered to be a promising solution since it can protect the copyright of multimedia data. However, until today, watermarking techniques are still far from practical. In author's point of view, it is quite difficult to make it commercial in the near future. As a result, although many watermarking companies exist at present, none of them seem profitable for the moment.

1.2 Applications

Digital watermarking is being used in numerous applications. Most of the current applications are devoted to copyright protection. It has the following purposes in general:

(1) Covert Communications: These are mainly applications of steganography. In military and intelligence applications, people would like to send messages to each other without being detected.

(2) Authentication: Sometimes it is necessary to verify the authenticity of input data, i.e., to determine whether the data are original, fake, or the altered version of the original. For authentication purposes, fragile watermarks seem to be a good solution; a properly designed fragile watermarking algorithm should be able to detect any alterations.

(3) Identification of Ownership: Robust watermarking algorithms are developed to identify the ownership of digital media. In this kind of applications, a movie producer selling its products in digital formats is subject to copyright piracy. In such situations, original producers would like to have legally proof that they are the real owners. A well-designed robust watermarking scheme is a possible solution to these cases. Additionally, to prevent the unauthorized users from playing the digital products, some media player manufacturers consider adding the watermarking detection facilities in their products. In such a scenario, media players would play the input clips only if they successfully detect the watermark of the company and confirm the authorization of the consumers. This is similar to DRM (short for digital right management) solutions except that watermarking is more resistant to media processing.

Classification of Digital Watermarking Applications

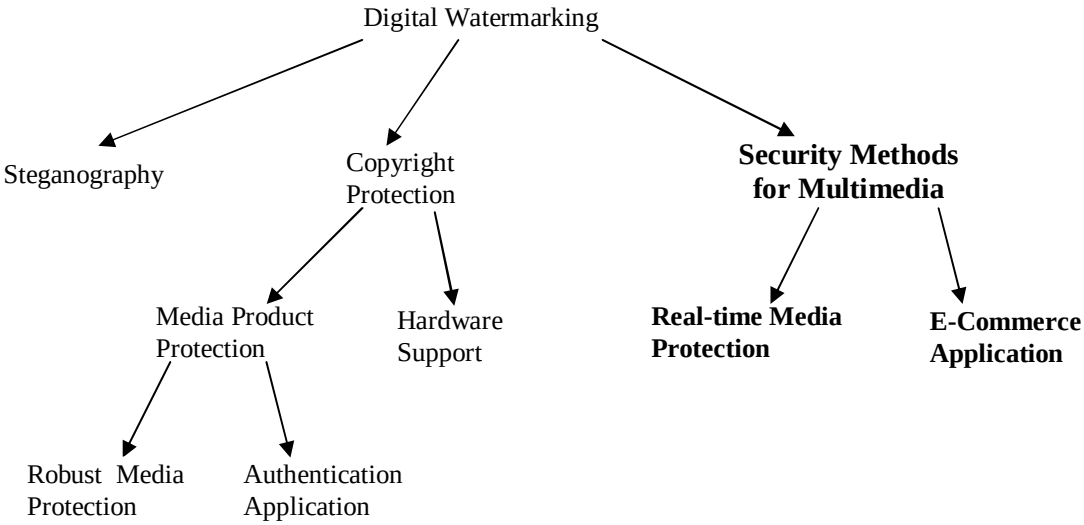


Figure 1.1: Classification of Digital Watermarking Applications

Different scenarios for digital watermarking yield different constraints and specifications. Therefore, they have to be treated differently. The watermarking applications can be categorize according to their usage and requirements. Figure 1.1 illustrates our classification.

The first class is *steganography*. Here the goal is to have a covert communication channel between two parties of the communication. The existence of this channel should be unknown to a possible attacker.

The second class is *copyright protection*. Here, the goal is to embed copyright information within the host data. The possible hidden data is known at the receiving end. The receiver is designed to "verify" reliably the existence (or non-existence) of a digital watermark in the received data. It has three kinds of applications:

- *Robust media protection.* The goal is to remain a digital watermark even undergoing some serious manipulation.
- *Authentication application,* its goal is to embed information in a robust way such that in case of manipulations, it should be possible to state that the data have been modified. Therefore, for fragile watermarking, broken watermark indicates the alterations.

- *Hardware Support.* It is the multimedia reader/writer accessorized with copyright protection components. The hardware solution is the ultimate way to eliminate the copyright piracy. The ideal solution is to install a small chip, which can recognize the copyright of media products, on DVD players in order to prevent the customers from copying commercial videos.

The third class is *security methods for real-time multimedia communication*. Here, the hidden message is known to both the embedder and receiver. If the original host data are available (respectively unavailable) at the decoder, then the problem is *blind decoding* (respectively *non-blind decoding*).

Some researchers are working on applying digital watermarking into multimedia data indexing [1.2, 1.3].

1.3 Core of the Thesis

Digital watermarking can link some useful information to the multimedia data by embedding watermarks into the original data. Hence, an attacker cannot remove the embedded watermarks easily. A large variety of digital watermarking algorithms has been proposed and nearly none of them is applicable for real-life applications. Many of these algorithms operate in time domain or frequency domain.

The digital watermarks embedded in the original data usually include some useful information, e.g., name of the authors, date of generation, or copyright holders. With the use of the blind digital watermarking algorithm, the embedded digital watermark can be extracted accurately without the need of the original multimedia data. Fragile audio watermarking algorithms can detect severe tampers/attacks occurred on the multimedia data. Therefore it is a useful method to the ensure authenticity and integrity of multimedia data.

Fragile digital watermarking provides an alternative approach to ensure the safety of multimedia data during transmission in openly accessible channels. That is, digital watermarks may be generated referring to information on originators, receivers, unique serial number, and time stamps. These watermarks are then embedded into the multimedia data to assure its integrity and origin source authentication without degrading the overall quality of the transmitted multimedia data.

When inspecting the integrity and authenticity of the transmitted multimedia data, some special characters should be taken into account--- the amount of the multimedia data is unpredictable, and the occurrences of packet loss and bit errors are also unpredictable. Figure 1.2 illustrates the outline of a security mechanism using digital watermarks. Digital watermarking operates on the multimedia data to hide/extract information and this makes

this approach different to most of the current cryptography mechanisms. Since the receiver can only receive the unpredictable multimedia data, we need an independent reference with the help of which to verify the integrity and authenticity of the dynamic multimedia data in the transmission. In the proposed approach, both the transmitter and receiver share one reference watermark from an independent trustworthy party before the multimedia data transmission really starts. To provide the non-repudiate and authenticity, a secret digital watermark is deployed and the secret digital watermark has been encrypted with public key algorithm, i.e., RSA, for the purpose of source origin authentication.

The transmitter embeds the digital watermark, *W*, into the compressed data stream, *M*, to form the outgoing multimedia stream, *M'*. The receiver extracts the embedded digital watermark, *W'*, from its incoming multimedia stream, *M"*. This scheme is an integrated solution to secure multimedia communication and multimedia data integrity.

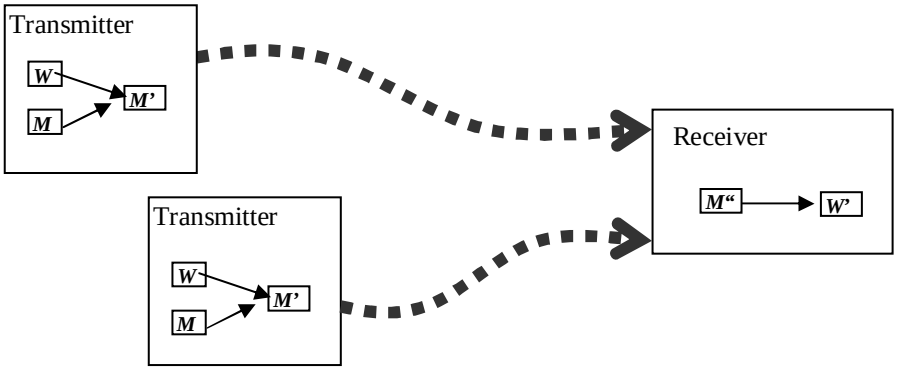


Figure 1.2: A Secure Real-time Communication Integrating Digital Watermarking

The network transportation can be viewed as a noisy channel. The reference digital watermark is modulated with multimedia data (carrier) and transmitted onto the noisy channel. The digital watermark undergoes the same alterations suffered by the multimedia data, thus that the watermark degradation can be used to estimate the overall alterations of the multimedia data caused by noises/attacks. So on the receiver's side, the embedded digital watermark is extracted, and compared with the original reference watermark to measure the integrity and authenticity of the received multimedia data.

In Internet telephony and video systems, an audio/visual data packet contains the compressed sample for 20 ms or 50 ms, thus a packet loss does not affect the session too much. Integrity and authenticity methods (i.e., SRTP [2.1]) which deploys Hash functions can measure the integrity of each data packet instead of the whole conversation data which consisting of hundreds or thousands of audio data packets. In this thesis, an alternative integrity measurement intended for a whole conversation is presented.

1.3 Outline of the thesis

In Chapter 2, we first introduce the security methods deployed in current multimedia communications and then present a novel approach to secure the multimedia transmission problem using digital watermarking techniques. The proposed scheme has three advantages: 1, the computation consumption is rather low; 2, source origin authentication becomes possible; 3. security scheme protect the whole contents instead of one packet.

In Chapter 3, we employ the theory developed in Chapter 2 on secure image transmission. The proposed security method can be implemented by the low-cost, low-resource front-end devices.

In Chapter 4, a speech authentication scheme based on SVD decomposition is proposed first; second, a speech watermarking algorithm using frequency domain linear prediction is introduced; third, another watermarking algorithm using analysis/synthesis model is presented; at last we present a fragile speech watermarking algorithm integrated with GSM 610 voice encoder. The last speech watermarking algorithm slightly modifies some niches of RPE pulse values; hence it does not degrade the overall quality of speech.

In Chapter 5, we further focus more on security issues of VoIP systems. Here, we present the security scheme using speech watermarking technique. The security framework with use of digital watermarking can meet the requirements of integrity and source origin authentication in a VoIP scenario. In addition, the timing mechanism is integrated by a quite natural manner.

In Chapter 6, we give an application in e-commerce with the use of the technique presented in Chapter 5.

Chapter 2 Applying Digital Watermarking for Networked Multimedia

2.1 Content Security of Networked Multimedia Communication

In networked environments the safety of multimedia data can be investigated according to two aspects — the *safety* of static data and the data *security* during dynamic communication. The safety of static multimedia data can be inspected according to the following four aspects:

Storage: Is the data centrally stored, or dispersed?

Vulnerability: How robustness is the data against to theft or abuse?

Confidence/Authenticity: What constitutes authentic information? Can that information be tampered with?

Linking: Will the multimedia data be linked to other information, e.g., about originating and/or consuming party?

When inspecting the security of real-time multimedia communication, one should take into account the specific properties of both multimedia data and real-time communication. First, limited distortions in multimedia data can not be perceived by end users, so some bit errors and packets loss that may occur during communication do not defect the overall visual/audio quality. Secondly, due to scheduling protocols of real-time multimedia communication, packet loss may happen. Thirdly, caused by the large amount of multimedia data, communication security trade-offs should be low enough.

Many kinds of cryptographic protocols have been proposed for the security of real-time multimedia communication. On one hand, some traditional cryptographic protocols, which were originated as dedicated mechanisms for Web security, were adapted to real-time multimedia communication, e. g., PGP [2.14] is being exploited by Speakfreely [5.8]. On the other hand, some security protocols specific to multimedia communication were developed, e.g., SRTP [2.1] and H.235 [2.10] for IP telephony. Most of these cryptographic protocols provide the functionalities of encryption, integrity, and authenticity. These security aspects can be provided by means of different cryptographic techniques such as secret key cryptography, public key cryptography, and hash functions, respectively.

The first half of this chapter introduces some well-known generic methods to secure multimedia data. These approaches are, however, not directly suited to solve the authentication problem for multimedia data transmission under the constraints of real-time operation and of the considerable restrictions of many low-end devices being used currently in ubiquitous computing scenarios.

Multimedia application systems can be divided into two categories: Stored Multimedia Data Stream and Real-Time Multimedia Interactive. A stored multimedia application involves some forms of interaction (action-reaction, request-response or exchange of information) between two parties (i.e., web browser-to-web server, ftp or http). Examples of interactive applications include the following: IP telephony, interactive voice/video, videoconferencing, Video-on-demand (VOD), and virtual reality.

The elapsed time between interactions is essential to the success of an interactive application. The degree of interactivity determines the level or stringency of the delay requirement. For example, interactive voice applications, which involve human interaction (conversation) in real time, have strict delay requirements (in the order of milliseconds). Streaming (playback) video applications involve less interaction, (i.e., interaction mostly occurs during start, stop, forward, or reverse action on the video) and do not require real-time response. Therefore, streaming applications have more relaxed delay requirements (in the order of seconds). Often the applications' delay tolerance is determined by the users' delay tolerance (i.e., higher delay tolerance leads to more relaxed delay requirements).

2.1.1 Stored Multimedia Data Stream

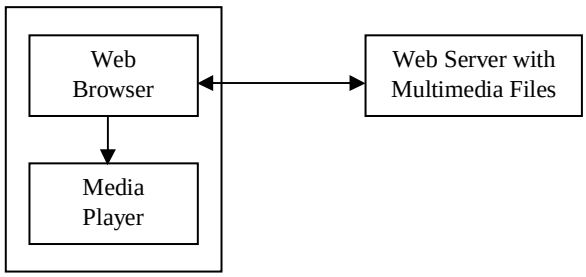


Figure 2.1: Simplest Stored Multimedia Approach

The simplest stored multimedia approach is shown as Figure. 2.1. The most commonly-used applications include:

- Extraction of audio or video stored in file
- Multimedia files transferred as HTTP object
- Multimedia mail
- Multimedia notes

In this sub-section, we give a brief introduction to traditional cryptographic solutions for the stored multimedia applications, and present some techniques.

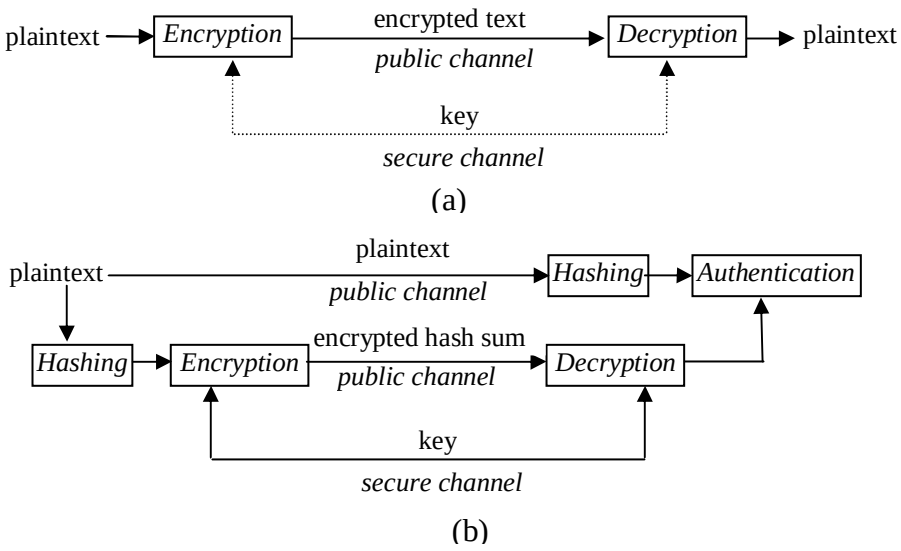


Figure 2.2: (a) A Cipher System; (2) A Digital Signature System

Encryption

Confidentiality, also called secrecy or privacy, is the most fundamental security requirement for data communication. It is usually achieved by means of symmetric encryption or asymmetric algorithm.

In a symmetric system [2.13], both encryption and decryption are performed with the shared key. In the early years, the participants use the shared key for multiple times. From the world war two, the one-time secret key has been widely used. The shared key is usually exchanged between the participants with the use of public-key encryption (or called the asymmetrical cipher system). The security of symmetric encryption is vulnerable since the attackers can use some skills, i.e., known-plain text attack and chosen-plain text attack to find the secret key being used.

In an asymmetric system [2.13], each participant has a key pair consisting of a public key and a private key. All participants publish their public keys. If one person, A, wants to send the other one, B, a message, he just encrypts the plain message with B's public key and sent the encrypted text to B. Only B than other else can decrypt the incoming text with B's private key. The robustness of the asymmetric system is rather strong. In the later part of the thesis, both the symmetric and asymmetric algorithms will be deployed in our implementation.

Message Authentication

On the contrary to secret encryption algorithms, message authentication algorithms [2.13] transmit a plain text and an authentication code together. Message authentication does not provide secrecy, but focuses on the integrity and non-repudiability of the message. Every one can read the plain text, but any alteration on the text can be noticed. Among many message authentication algorithms, some deploy asymmetric encryption algorithms to protect the plain text from forgery. Message secrecy is usually accomplished by encryption.

The transmitter and receiver share the same secret key to compute and verify the message authentication code. The message authentication code works as follow: 1) use a one-way hash function to produce a checksum; 2) encrypt the checksum and append to the message; 3) the recipient hashes the received message and compare the checksum with the received message authentication code.

Any alterations made to the message would completely alter the hash code.

2.1.2 Real-Time Multimedia Interactive

Real-time system is a system can support the execution of applications within a specified time window. The time constraints for external processes are seconds for flow processes, milliseconds for electric power systems, and 20ms~1m for video and audio streaming. Video and audio streaming provides the means of delivering news, entertainment, remote education, presentations, and many more types of live communication within a short time constraint. Television is the most well known application of streaming media. It already feeds wireless multimedia streams into millions of dishes and antennas, connected to TVs and other devices.

Streaming technologies are important since most users do not have access to enough connection capacity to download large multimedia files quickly. Using streaming technologies, consumers can start listening to the audio stream or view the video stream

before the entire file has been received. To allow efficient streaming, the provider needs to send the data as a steady stream and the receiver needs to be able to cache excess data in a temporary buffer until used.

To reduce the amount of information transmitted, streaming video and audio data are compressed by means of technologies such as MPEG. The streaming video quality depends on the capacity of the transmission channel and its ability to support a steady stream—the better the channel quality (i.e., higher and steady data rate), the better the quality of the audio and video output.

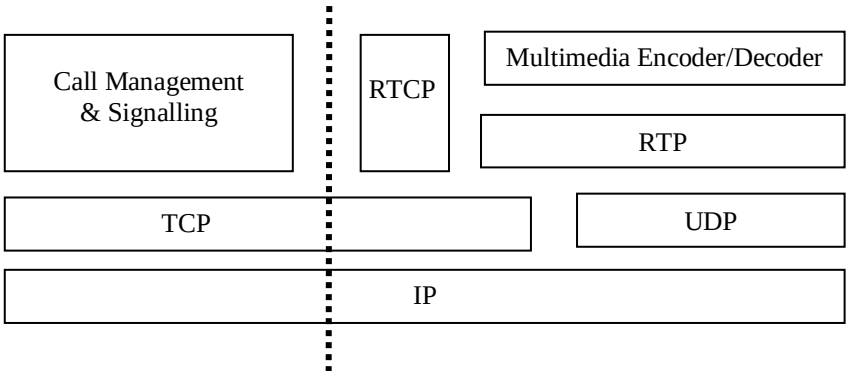


Figure 2.3: Streamed Multimedia Application Systems

Figure 2.3 illustrates a typical multimedia streaming system [2.8]. Standing on the top of UDP or TCP layer, the early TCP/IP architecture does not provide the necessary mechanisms to meet the time constrained requirements of real-time streaming. So IETF specified the Real-time Transport Protocol (RTP) [2.11]. RTP is the default standard for audio/video transport in IP networks, which was developed the Internet Engineering Task Force (IETF) as RFC 1889. RTP is a framework for audio or video data delivery, and it has some profiles for particular uses. Standing on the top of

The RTP Specification

RTP [2.11] is an application level protocol operating over UDP/IP protocol stacks or ATM networks. It supports 1 user-to-1 user communication and 1-to-many broadcast as well. RTP receives compressed multimedia data from upper audio/video encoder and feeds to UDP stack. RTP provides some useful functionality, including timing mechanisms, loss detection, quality reporting, for continuous multimedia streaming. There are two parts of RTP: the data transfer protocol and an associated control protocol. The RTP data transfer protocol deliver real-time multimedia data between two peers in networks. The RTP control protocol (RTCP)

is a signalling protocol providing reception quality feedback, participant identification, and synchronization between media streams. RTCP runs alongside RTP and provides periodic reporting of this information.

RTP Payload Formats

RTP payload formats specify how particular media types are carried within RTP data packet. The individual vendors propose its own proposal to IETF, i.e., RFC 2429/2190 for H.263 video RTP, RFC 2435 for JPEG compressed video, and RFC 2250 for MPEG1/MPEG2 Video.

Security

In real world, VoIP systems may suffer many kinds of security threats. This is caused by the openness of VoIP implementations since most of the VoIP are operated on public accessible networks. The hackers may attack the calling server, invade the operating system, embed the virus in the VoIP telephone end points and even perform the DoS attacks to the signalling systems. In this thesis, we will only focus on the content safety of the multimedia data.

The confidentiality of RTP can be implemented by means of encryption, and the encryption might be performed at the application level or at the IP level. Perkins analysed the advantages and disadvantages of the two options in his recently published book [2.8]. RTP standard does not provide integrity protection or source origin authentication for the packetized data by itself. The designers of RTP specification leave it blank on purpose, thus the different developers can implement the security methods in their needs. Most of the authentication implementation is with accordance to secure RTP [2.1] or with use of IPSec [2.2].

2.1.2.1 Security Methods in the RTP Specification

The RTP specification [2.11] designs the encryption procedures. The deployed scheme encrypts RTP packet or RTCP packets. Since RTP packets and RTCP packets are processed on the top of IP layer, the encryption of the whole RTP or TRCP packets does not affect the routing and traffic of real-time multimedia data communications. The block encryption algorithms, i.e., DES-CBC [2.12], or stream ciphers are most commonly used. Both the transmitter and receiver share the same key and they need to negotiate to generate a

key before the start of the real-time data communication. The distribution of the shared key relies on the key exchange protocols. Figure 2.4 illustrates an encrypted RTP data packet. The RTP data packet is encrypted on the application level of TCP/IP stacks.

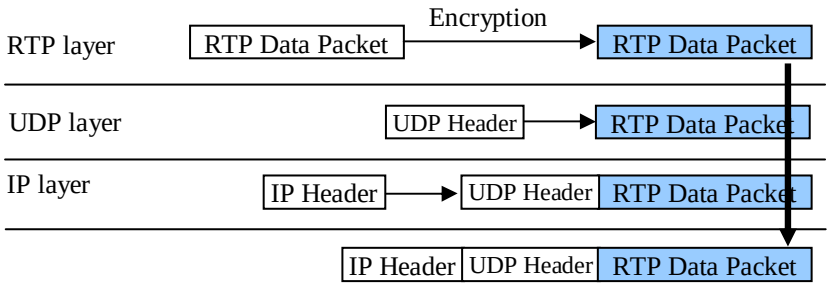


Figure 2.4: An Encrypted RTP Data Packet

DES-CBC is specified in both RTP specification and SRTP. Many VoIP systems being used presently are running on either a desktop or a notebook and thy implement the DES algorithm by a software module. DES is not robust enough by a software implementation, so the confidentiality specified by the RTP specification is vulnerable. In addition, the key distribution is also a point always to be attacked. RTP only authenticate the identification of participants, not the RTP data packet, thus RTP data packet is not difficult to be forged.

2.1.2.2 SRTP

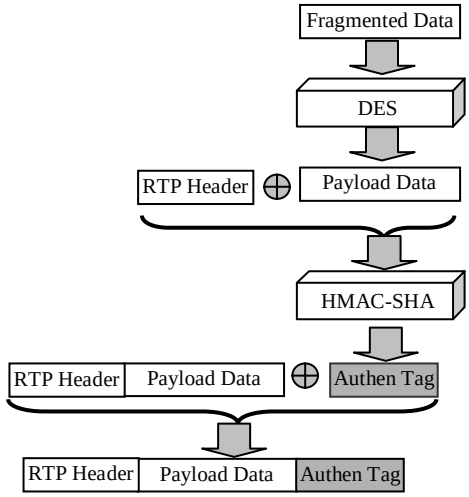


Figure 2.5: Integrity and Confidentiality Mechanic of SRTP

To overcome the security weakness of RTP specification, SRTP [2.1] is proposed by IETF. In SRTP, the transmitter and receiver need to use a same encryption algorithm and share the same keys. SRTP supports both message integrity protection and source origin authentication.

Source origin authentication can be achieved by means of message authentication code. TESLA (short for Timed Efficient Stream Loss-tolerant Authentication) is a promising algorithm specified by IETF as RFC 4082 [2.6].

The secure RTP specification describes the procedure generally as follow:

1) Data Encryption

$Payload\_Data_{encrypted} = DES(key_{DES}, Payload\_Data)$

2) Hashing

$Authen\_Tag = SHA(Payload\_Data_{encrypted})$

Authentication

As illustrated in Figure 2.5, message authentication code is used in SRTP to achieve the source origin authentication. The fragmented data is encrypted using DES algorithm and then an authentication code is generated by SHA algorithm over the whole data packet. Both the transmitter and receiver share a secret key to encrypt and authenticate the SRTP data packets.

Confidentiality

SRTP encrypts just the payload section of a RTP data packet to achieve the confidentiality, as shown in Figure 2.4. The specified encryption algorithm is DES-CBC [2.12]. The weakness of SRTP confidentiality solution is similar to the security method of RTP specification: the symmetric cipher based on software implementation is not robust enough.

2.1.2.3 IPSec

Authentication

IPSec [2.2] and SSL are two well-known solutions for the purpose of VPN security. SSL works in the application level, while IPSec operates on the network level and works just under the TCP/IP protocol stacks. IPSec [2.2] is a protocol set to provide encryption, integrity and authentication for data packets. It consists of a set of cryptographic protocols and Internet Key Exchange protocol (IKE for short).

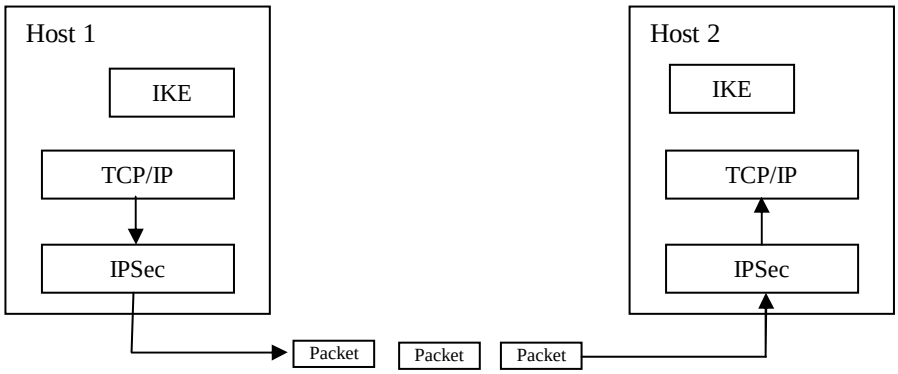


Figure 2.6: IPSec Architecture

Authentication in IPSec is provided by the Encapsulating Security Payload profile, and it encrypts the payload data and UDP header, and appends an authentication header and a trailer. Authentication Header (AH) [2.7] is another implementation choice standardized by IETF earlier. The key difference is that the entire packet is authenticated, including the outer IP header.

Confidentiality

ESP [2.2] also supports confidentiality of data packets. ESP adds an additional header and trailer to each data packet. A security parameter index and a sequence number are inserted between IP header and the payload data, and an authentication code is appended at the tail.

2.1.2.4 Fast Encryption Methods for Audiovisual Data Confidentiality

The existing security approaches work in different levels/components in secure multimedia communication systems. Selective algorithm [2.3], ZIG-ZAG permutation [2.4] and Video Encryption Algorithm [2.5] incorporate cryptographic techniques with multimedia compression algorithm. They are working in the application level.

Integrity and authenticity mechanisms of the existing security approaches have some drawbacks: they are not flexible enough to handle the distortion introduced during by communication; the receiver can retrieve the source multimedia data, and that may cause multimedia data abuse.

2.1.3 Drawbacks of Traditional Security Methods

Real-time multimedia communications unveils some special properties stemming from both multimedia data and real-time communication. First, end users cannot perceive limited distortions in multimedia data, so, some bit errors and packets losses occurring during communication do not defect the overall visual/audio quality. Secondly, due to controlling protocols and implementations of real-time multimedia communication, packet losses may happen any time. Thirdly, caused by the large amount of multimedia data, the communication security trade-offs should be as low as possible.

In the previous security methods, source origin authentication is achieved by hashing. However, hash function was originally developed as a solution to the integrity and authenticity problem for text messages. So, it does not take into consideration explicitly the special properties of multimedia data nor of real-time communication. The hashing is a one-time security method, thus it does not have any correlation with timing. Thus, there are some serious drawbacks when one attempts to exploit these methods for the envisaged application.

Firstly, hashing is sensitive to data distortion — an unrecoverable bit error in the multimedia message or in the digital signature/MAC code may disable the corresponding authentication procedures.

Secondly, in general, hashing-based authentication methods are facilitated on the checksums of message. These kinds of authentication methods append a pad at the tail of the multimedia data. The authentication pad is apart from the message itself. A possible attacker may discard the padding and record the multimedia contents only and then reuse this message again and again. The attackers can use the recorded voice to cheat the access control systems. In a scenario of multimedia news service or video on demand service, the attacks can record the contents of the TV programs or films, and resell the programs or films to the other customers for a profit.

Thirdly, these techniques increase the latency of multimedia communication so that the handheld devices (such as PDAs and cellular phones) in general cannot meet the resulting computing power requirements. With the incoming age of 3G wireless communication, multimedia news service, video conferencing will be realized on hand held devices such as PDAs and cell phones. All these multimedia services consume very large data volume. To achieve the privacy, the symmetric methods can be used. But as to the confidentiality, the asymmetric schemes are absolutely needed. The heavy processing of the asymmetric schemes may outbreak the computation capacity of portal devices. Thus an alternative solution with a light computation requirement is in urgent need.

Fourthly, hashing-based authentication techniques do not have any timing wrapping. That means that a series of authentication codes do not have any connection with timing, and the authentication codes sequence is orderless. It is difficulty to perform the post authentication based on the authentication code sequence.

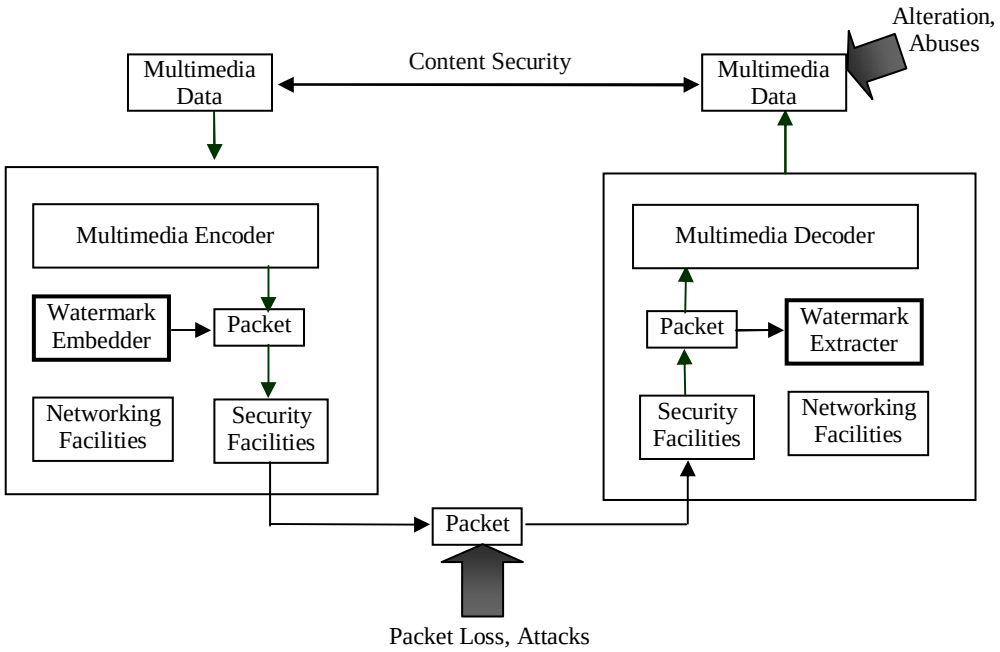


Figure 2.7: Content Security in Real-Time Multimedia Communication

Figure 2.7 illustrates attacks to the content security of real-time multimedia communication. During the course of transmission, multimedia traffic is subject to packet loss and malicious attacks. The receiver assembles the incoming data packets to restore the whole multimedia message. But the assembly process only extracts the data payloads from the packets and discards the packet header / tails which may contain the authentication codes or timing information. Thus the saved multimedia data does not have any authentication code linked to it. Thus the received multimedia data is subject to the potential tampers or abuses.

The possible solution is to embed the timing information and authentication codes into the multimedia data thus that the assembly process cannot detaching timing and authentication codes from the multimedia data. In the proposed scheme, the transmitter inserts some watermarks into the data packets and the receiver can retrieve the digital watermarks during the transmission to verify the security of transmission. The other parties can extract the watermarks from the stored multimedia data and perform the integrity and source origin

authentications. The utmost requirement of the digital watermarking algorithm is to detect any alterations.

2.2 A Watermarking Framework for Real-time Multimedia Communication

Some researchers [2.9] model the digital watermarking as a kind of communication problem. The technique of digital watermarking can be viewed as modulating a weak noise with a strong signal. If the noise is below a predefined threshold (HVS is a threshold for image watermarking and HAS is a reference measurement for audio watermarking), the distortion introduced by digital watermark can not be noticed by the person.

A typical watermarking framework is first proposed by I. Cox [2.9], as illustrated in Figure 2.8. A watermark embedder inserts a digital watermark, *w*, to an original digital media, *Do*. The watermarked media, D_1, should be perceptually identical to the original. The watermarked media may suffer various kinds of attacks during the distribution or transmission. A watermark detector extracts the hidden data, *w'*, from the received copy, D_2. The digital watermark may be a message, an image or a video clip. The model is originated from the well known communication model.

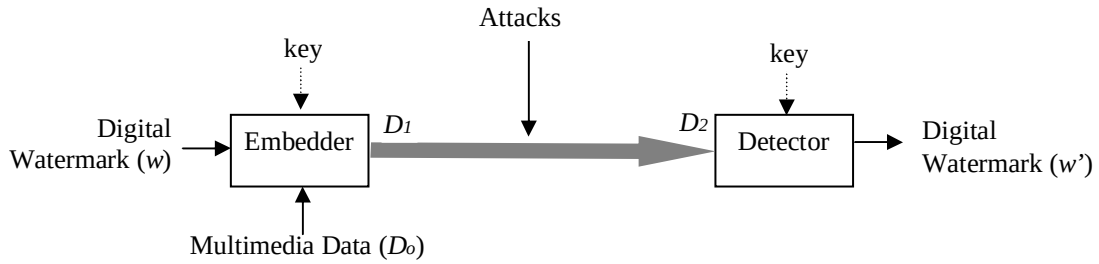


Figure 2.8: General framework of data hiding systems

2.2.1 Outline

Digital watermarking provides an alternative approach to ensure the safety of multimedia data during transmission in openly accessible channels. The digital watermarks may be embedded into the multimedia data to assure its integrity and authenticity without degrading the overall quality of the transmitted data.

Figure 2.9 shows the outline of a security mechanism using digital watermarks. Digital watermarking operates on the multimedia data to hide/extract information. This makes this approach different to most of the current cryptography mechanisms.

When inspecting the integrity and authenticity of the transmitted multimedia data, some special characteristics are to be taken into account - both the amount of the multimedia data and the occurrences of packet loss as well as of bit errors are unpredictable. Since the receiver can only receive the unpredictable multimedia data, we need an independent reference in order to verify the integrity and authenticity of the dynamic multimedia data during transmission. In the proposed approach, the sender and receiver share one reference watermark, w_1, from an independent trustworthy party before the multimedia data transmission really starts. To provide the non-repudiate and authenticity, a secret digital watermark, w_2, is being introduced. w_2 has previously been encrypted with a public key algorithm, e.g., RSA.

The sender embeds both a public, *w*1, and a secret watermark, *w2*, into the outgoing multimedia data stream. The receiver then extracts the public digital watermark, *w*1', from its incoming stream and compare *w*1' with *w*1 to determine the security of the session. The secret watermark, *w2*, can be used for the purpose of source origin authentication. This scheme provides an integrated solution to secure multimedia communication and multimedia data integrity.

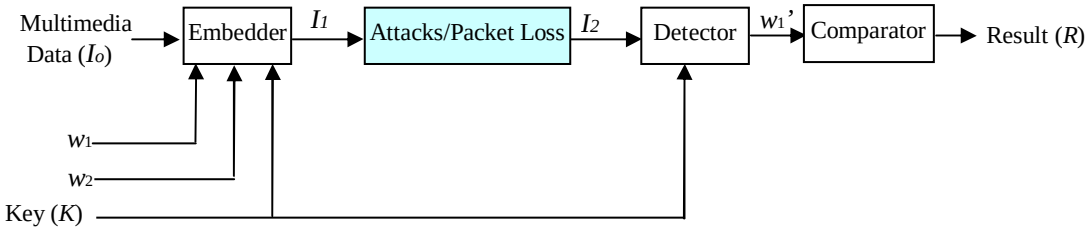


Figure 2.9: Secure Real-time Communication with Digital Watermarking

The network transportation path can be viewed as a noisy channel. The reference digital watermark is modulated with multimedia data (carrier) and transmitted onto the noisy channel. The watermark undergoes the same changes suffered by the multimedia data, so that the watermark degradation can be used to estimate the overall alterations of the multimedia data caused by noise or by attacks. At the receiver side the embedded digital watermark is extracted and compared to the original reference watermark in order to measure the integrity and authenticity of the received multimedia data. It is obvious that the proposed framework integrate the timing mechanism to the multimedia data naturally.

We can compare the proposed scheme with the traditional real-time multimedia authentication scheme, as shown in Figure 2.10. On the transmitter's side, in traditional

authentication scheme, the timing information is included in the RTP header and the source origin information is included in the appended authentication code. While in the proposed scheme, the timing information and source origin information are integrated into the digital watermark.

In the traditional scheme, the receiver extracts the data payload from the RTP data packet, and assembles multiple data payloads to form the recorded multimedia data. Since the RTP header and the appended authentication tag have been discarded, the assembled multimedia data does not contain timing information and source origin information any more. Appending a padding including the timing and source information would be a potential solution. However, the padding is independent of the protected multimedia data thus the saved multimedia data may be abused or attacked. The process of padding generation is also not trustable.

In the proposed scheme, the transmitter embeds the digital watermark into the data payload. By careful design, the receiver cannot eliminate the embedded watermark. The saved multimedia data contains the timing information and source origin information by itself. The watermark embedding and extraction are performed in the application layer, so the traditional security methods also can be employed to protect the real time communications.

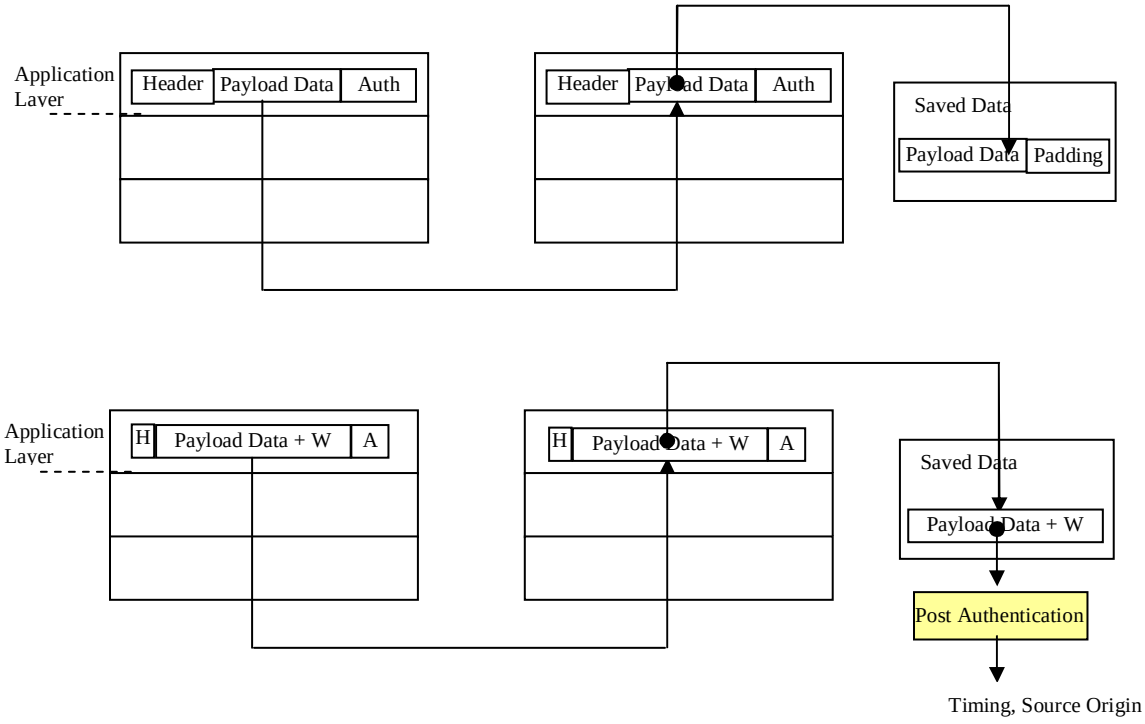


Figure 2.10: Secure Real-time Communication with Digital Watermarking

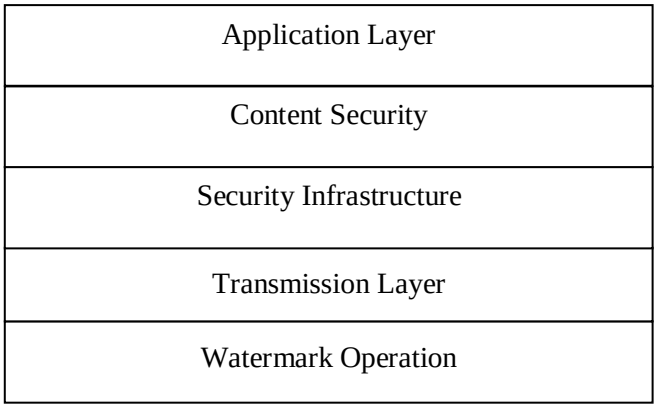


Figure 2.11: Layered structure of a data hiding system in real-time multimedia system

A layered structure of the proposed security framework using digital watermarking technique is illustrated in Figure 2.11. The "Watermark Operation" embeds watermark bits in the host signal or extracts watermark bit from the watermarked data. The "Transmission Layer" deals with the packet loss which may occur during the transmission and the bit error for stored media. The "Security Infrastructure" provides the traditional cryptographic service for the framework, and it may be implemented by SSL. The cryptographic service includes secure channel management and digital certificate authentication. The "Content Security" handles the security of the whole multimedia session to protect multimedia data from reuse and abuse. The "Application Layer" extends the content security, and one potential application is "Voice Cheque" presented in Chapter 6.

2.2.2 Unique ID and Transaction ID Assignment

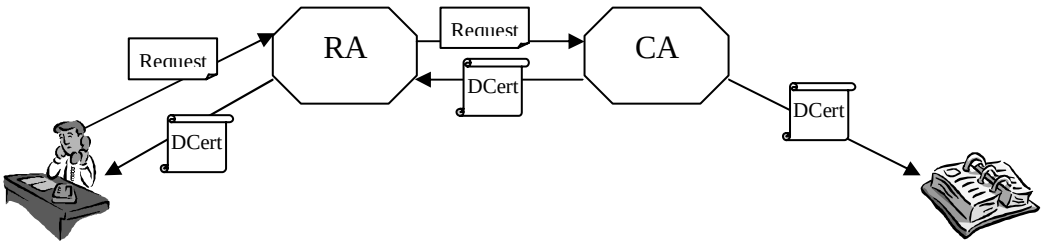


Figure 2.12: Linking a unique Certificate to each Device

In our work, each front-end device (fingerprint scanner, cellular phone or Internet telephone) needs a mechanism to link itself to a unique identifier, *DCert*, for authentication purposes.

A digital certificate is a means to authenticate identity. A certification authority (CA) usually utilizes a public key infrastructure (PKI) to generate, authenticate and manage the digital certificates. A digital certificate is usually issued by a trusted third party (CA) to bind an entity to a public key. The digital certificate is signed by the CA with the CA's private key. This provides independent confirmation of an entity.

SSL is being used in the proposed approach to generate the required unique identifiers. At start, a device combines some information such as its name, IP address, telephone number, and email address to generate a digital certificate request. Then the device posts the digital certificate request to a register authority (RA). The RA verifies the incoming request and forwards it to a certificate authority (CA). The CA issues a digital certificate and sends then back the certificate to the applying device. In addition, it stores the issued digital certificate into the certificate directory.

A kind of transaction certificates denoted *transCert*s are assigned to each conversation in order to uniquely identify a transaction. A transaction server denoted as *transServer* generates the *transCert*s, and distributes them to the participating front-end devices, i.e. fingerprint scanner, cellular phones or voice over IP partners, respectively.

The generation of a *transCert* is illustrated in Figure.2.13. In its simplest form, a *transCert* would include the identifier of the individual/entity, timestamp, serial number for the certificate, and the individuals' or entities' public key. Most importantly, it would be digitally signed by the issuing *transServer* using the *transServer*'s private key. A *transCert* could contain other information as well. Depending on the type of certificate, it could include information on access privileges, geographic location of the owner, or the age of the owner.

First, *transServer* generates a message packet consisting of a timestamp, identifiers of participants, transaction number, and a random number. Secondly, *transServer* signs the message packet using a public-key encryption algorithm to create a secret watermark, denoted as *w*2. Third, *transServer* creates a *key* and a public watermark, denoted as *w*1, respectively.

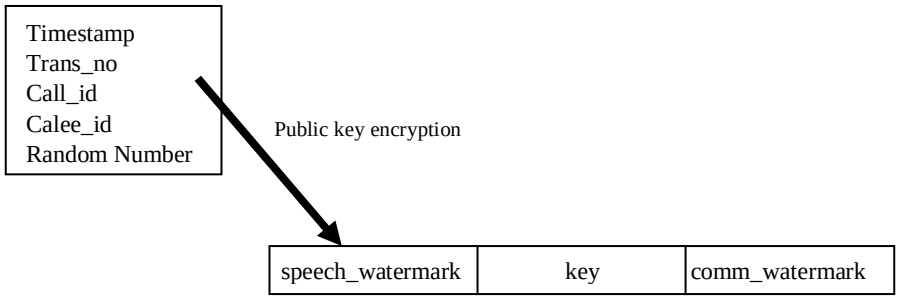


Figure 2.13: An Illustration of *transCert* Generation

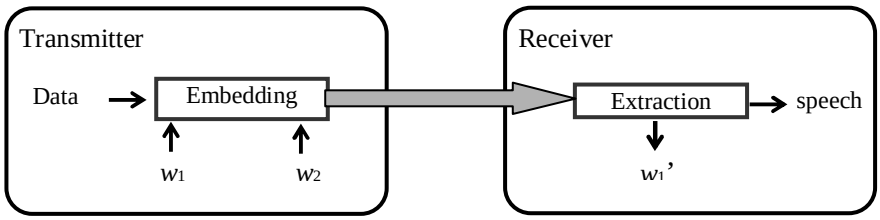


Figure 2.14: Real-Time Authentication for VoIP

Once the participants receive the *transCert*s from *transServer*, the conversation may begin. At first, they authenticate the received transaction certificate. If the authentication is successful, then the participants parse the received *transCert*s to ((*w*2, *key*) *w*1) and insert then the digital watermarks into their outgoing audio data streams. At the same time, both participants extract the *w*1*'* from their incoming streams and authenticate them, as depicted in Figure 2.14. At the end of the transmission, i.e., conversation, each of the participants sends a *dialog_end* request to *transServer*. *transServer* authenticates the received *dialog_end* request and puts them on file.

2.2.3 transServer

transServer has a facility of CA. The certificates, *transCert*s and *DCert*s are digitally signed by *transServer* providing independent confirmation that entities or transactions are in fact who they claim to be and that the public key provided by them does in fact belong to that party. The *transServer*'s public key is widely known so that there is no need to authenticate

transServer's digital signature. Each devices or end user is relying on the *transServer*'s digital signature to authenticate the certificate owner's identity and to bind that identity to their own public key. The procedure can be formulated as:

answer ← Digital_Cert_Authentication*(transCert* or *DCert*, public key *)*

2.2.4 Source Origin Authentication

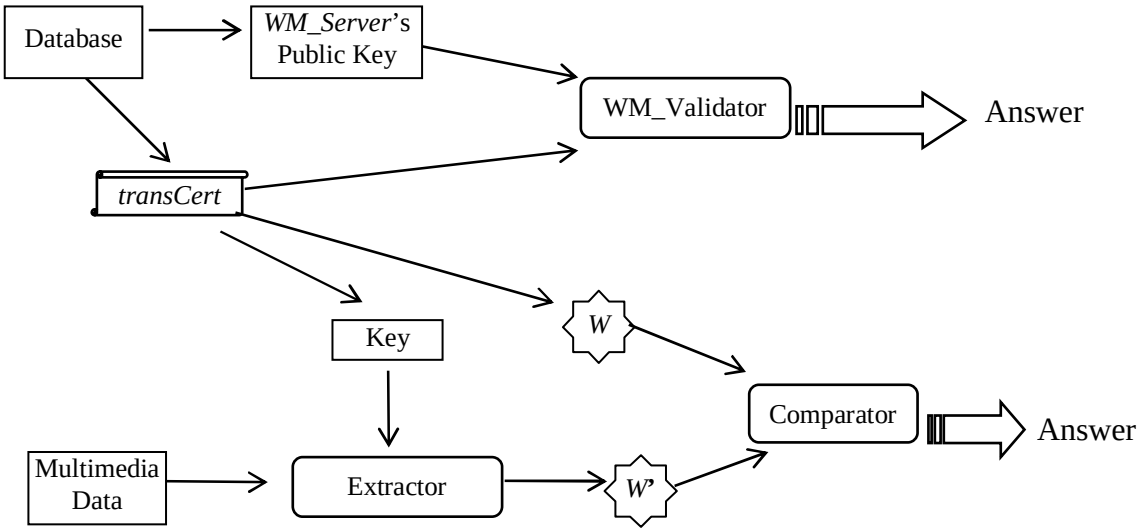


Figure 2.15: Source Origin Authentication

The digital watermarks embedded in the multimedia messages have been signed using public-key encryption. This means that each of the watermarked multimedia data is distinguishable from others. Therefore, one can verify the source origin of the saved multimedia data as follows:

1) parse a *transCert* to extract the proper digital watermark, *w*

   (*w, key*) ← Parse(*transCert*)

2) extract a digital watermark, *w*', from the recorded multimedia data

   w' ← Extract(*data, key*)

3) compare it with the original one to authenticate the inspected multimedia data.

   answer ← (*w* ≡ *w*':?)

Post authentication as illustrated in Figure 2.15 is intended to be used by anyone. A suspected image can be authenticated by assessing the similarity between, *w*', the embedded watermark and, *w,* reference watermark.

Since *w* has initially been signed by means of the RSA private key of the authentication server, *transServer*, the validation of *w* can be authenticated by applying a RSA encryption operation with use of *transServer*'s RSA public key. The process can be formulated as:

answer ← RSA_Authentication(*Server's public key, transCert*)

Post authentication is performed by some auxiliary tools denoted as *MM_Authenticator* and *WM_Validator*, respectively. *MM_Authenticator* extracts the digital watermark, *w'*, from the watermarked multimedia data and then compares it to the original watermark, *w*, in order to make an assessment of authenticity. In contrast, *WM_Validator* is used to authenticate the author of *w*. Anyone can use these tools in order to authenticate suspected multimedia data and *w*. Finally, one characteristic property of the proposed approach should be mentioned - both *w* and *transServer*'s public RSA key are accessible to anybody. So, both image and watermark authentications are public.

Post authentication is widely used in E-commerce and email systems. In e-commerce, the post authentication enables the participators of the e-transaction to validate the data sheet after the transaction; in email system, the post authentication can verify the source origin and integrity of the text or multimedia message. For real-time multimedia systems, the proposed post authentication scheme can verify the source origin, integrity, timing and source origin of the real time multimedia data. This can enable the e-commerce applications with more natural human computer interfaces by replacing the text-based data sheet with multimedia data such as speech or video. Chapter 6 introduces a possible e-commerce application scenario.

2.2.5 Application Areas

In this work, some security mechanisms are exploited, which rely on authentication rather than on encryption methods. The highly asymmetric architectures found in ubiquitous computing applications are exploited to provide a protection of the transmitted multimedia data by means of well-known digital watermarking techniques. The advocated approach adopts public key encryption to efficiently generate non-repudiate multimedia data, whereas both the digital watermark generation and the authentication algorithms are public. In addition, this approach to real-time multimedia data authentication is independent of specific network properties, so it may easily be exploited for heterogeneous communication networks. With the use of the proposed integrity and authenticity mechanisms, we link some useful information to the multimedia data and as well can detect any tampers occurred on it. The

proposed security scheme can be applied in different scenarios such as deploying static image watermarking techniques in multimedia collection and applying audio watermarking techniques in e-commerce scenarios.

To prove the evidence of the proposed scheme, a prototype for secure digital image communication has been implemented. An improved image watermarking algorithm is exploited to embed digital watermark into image DWT coefficients.

In addition, another prototype for secure speech communication has been developed and implemented. The prototype architecture consists of a pair of IP phones and a *transServer* which is a software module running on a powerful computer with a RSA facility. A new fragile speech watermarking algorithm incorporating with GSM 610 full-rate voice encoder is deployed to detect the tampers.

Applying Digital Watermarking Techniques for Secure Image Capture

Front-ends of image collection systems are in general to be implemented by means of low-cost, low-resource devices. Since this kind of devices embedded into the authentication application feature in most cases a weak computation power only, a straight forward security policies such as encryption is not a good choice. We exploited digital image watermarking technique to this asymmetric computing architecture.

At the front-end´s side, the complete image is first divided into a sequence of images clips { I_0, I_1, I_2, …, I_M }. The digital watermark, W, can also be structured into a sequence of sub-watermarks, { W_0, W_1, W_2,…, W_M }. Each sub-watermark is then to be inserted into the corresponding image clip in order to create a watermarked clip. Then the watermarked image clips are sequentially transmitted to the server, which is in charge of image reconstruction and of digital watermark authentication. Finally, the watermark detector of the server extracts a sequence of sub-watermarks denoted as { W_0**'**, W_1**'**,…, W_M**'** } from the received set of marked image clips and decides on the authenticity of the transmitted image. In order to ensure the overall safety of the communication, real-time authentication is employed during the transmission. In the beginning of each transmission cycle, *Watermark Embedder* on front-end inserts a digital watermark, W_i, into the i-th clip of the original image, and then the watermarked clip is sent to *Server*. As soon as *Server* receives the watermarked image clip, it extracts the embedded watermark, W_i', immediately. *Server* compares the extracted watermark *W'* with the original one, *W*. If W_i' does not differ from W_i, then the received image clips are composed into the authentified final image. After accepting all watermarked

image clips, *WM_Server* reconstructs the full image and checks the authenticity of the received image.

The image watermarking procedure used here is operated in Wavelet Domain. Since DWT is very suitable for low-end chips, the image clip is transformed into wavelet frequency domain first, and then some coefficients are modified slightly to embed the digital watermark into it.

Applying Digital Watermarking Techniques for Secure VoIP

VoIP or Internet telephony is a technology allows people to perform telephone service using the existing IP-based data network, thus it can avoid the tolls charged by telephone service provider. However, lot of security incidents occurred in the last years and their number is still increasing. Therefore, security is a demanding topic for VoIP telephony systems.

There are five different categories of security services present for IP telephony: Identification and authentication, Authorization, Confidentiality, Integrity, Non-denial/Non-repudiation. These security aspects can be provided by means of different cryptographic techniques such as secret key cryptography, public key cryptography, and hash functions, respectively. In this thesis, we mainly address integrity and source origin authentication of multimedia data. By multimedia data integrity and authenticity mechanisms, we can provide some very useful related information, such as the speaker identity, the point in time when dialog happened, and the integrity/authenticity of the transaction.

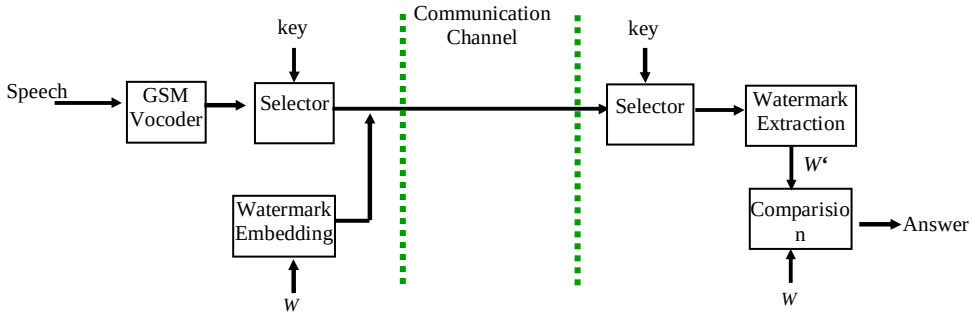


Figure 2.16: Application in Real-Time Speech Communication

In our work, two digital watermarks, namely *speech_watermark* and *comm_watermark*, are used for each transaction. For the purpose of source origin authentication and integrity, a speech message will be inserted into a unique fragile

watermark, *speech_watermark*. In contrast, a *comm_watermark* will be inserted into the speech message to ensure the authenticity and integrity of real-time speech transmission. Both digital watermarks and private keys are binary sequences.

This proposed scheme has several advantage and is suitable for real-time multimedia communication: 1, the proposed scheme is robust to packet loss and delays; 2, the watermarking procedure is lighter than encryption; 3, the proposed scheme provides integrity and authenticity functionality for the whole multimedia message instead of single data packet; 4, adopts public key encryption to efficiently generate non-repudiate multimedia data. In addition, this approach to real-time multimedia data authentication is independent of specific network properties, so it may easily be exploited for heterogeneous communication networks.

Chapter 3 A Digital Watermarking-Based Image Authentication Scheme for Distributed Embedded Systems

This chapter outlines a novel approach to extending well-known digital watermarking techniques to the protection of image data during the network transmission within distributed highly asymmetric system architectures. An incremental watermarking algorithm suitable for low-resource embedded systems to be used in distributed environments is presented. The proposed watermarked image authentication scheme is public; its safety depends on the robustness of the underlying RSA algorithm. The main advantages of this approach are demonstrated by some use cases.

3.1. Introduction

The upcoming information processing architectures for ubiquitous computing is highly sensitive to security issues. For some networked scenarios, such as fingerprint collection in distributed environment, video monitoring and health care systems, the image integrity and authenticity is fatal to the success of these services. While most of the embedded systems working in such distributed environments are low-end devices in terms of their computing power, memory size, and communication bandwidths. Therefore, new security policies have to provide such that the given constraints of these devices are considered accordingly.

When inspecting the security of real-time multimedia communication, one should take into account the special properties of both multimedia data and real-time communication. First, the limited distortions in multimedia do not disturb the end users or degrade the performance of biometric recognition systems. Second, packet loss and bit errors may always happen, especially for wireless connection. Third, due to the large amount of multimedia data, the communication security trade-offs should be low enough.

The existing integrity and authenticity approaches such as MAC (Message Authentication Code) have some drawbacks: first, they are sensitive to data distortion — an unrecoverable bit error in the multimedia message or the possible packet loss in real-time multimedia transmission may disable the corresponding authentication procedures; second, digital digest/MAC is apart from the message itself and this may cause multimedia data abuse; third, these techniques are packet-based such that not applicable to the continuous multimedia.

To overcome the drawbacks of existing solutions, some new security mechanisms using digital watermarking [3.1-3.7] techniques are proposed, which rely on authentication rather than on encryption methods. In this chapter, we report our work on applying digital watermarking techniques to enhance the security of image transmission. The main contributions outlined in the following are an incremental approach to watermark generation and a secure software architecture suited for highly asymmetric distributed identification systems.

3.2 Fragile Image Watermarking

3.2.1 Wavelet Method

Wavelet theory [3.18] provides a method to analyze and represents signals in time for its frequency contents. Wavelets are localized waves. A "mother function" or "wavelet basis", Ψ(t), is defined first. Then one can scale and translate the mother wavelet [3.18] to obtain a family of other wavelets, which can be defined as

$$\Psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-a}{b}\right) \tag{3.1}$$

where, *a* locates the position of the wavelet and *b* indicates the width of wavelet. In a wavelet transform, any finite energy signal can be represented by a linear combination of wavelets Ψ(a,b)(t).

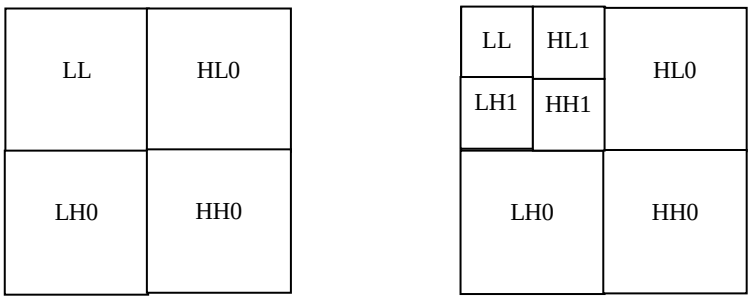


Figure 3.1: A Two Layered Wavelet Transformation

Since P. Goupillaud and J. Morlet [3.22] introduced the technology as a mathematical tool in 1980s, wavelet theory has been widely applied in physics and mathematics. In the last fifteen years, the technology is quickly adopted in image or video processing. The 2-D wavelet transform [3.18] is the basis for image processing. It can be achieved by performing

the 1-D DWT separately on rows and columns of an image. The 2-D wavelet transform produces four sub-bands, namely LL, LH, HL and HH. The LL sub-band represents the original signal in half resolution and contains smooth spatial data. The HH sub-band consists of noise and the edges in the image. Most of the image energy is concentrated in the LL sub-band. By applying the 2-D transform to the LL sub-band of the current scale repeatedly, a hierarchy tree of wavelet coefficients can be obtained. The benefit of wavelet transform lies in fitting in human visual system quite well. For the wavelet tree representation, the skeleton of an image has a low frequency, while the details have a high frequency. Since the digital watermarking must restore the image or audio after embedding signatures into the coefficients, only the wavelets with the property of reconstruction-able and discrete transformation can be used in watermarking. The commonly-used wavelets include: Meyer wavelet, Haar wavelet, Daubechies Wavelets, Symlets, Coiflets, Biorthogonal wavelets [3.19].

Usually the watermark embedding stage is invoked after DWT and quantization, and prior to entropy coding. Authentication is carried out in a reverse manner. In this work, the 'Daubechies db1' wavelet base is used for DWT. Since DWT is very suitable for low-end chips, the image clip is transformed into wavelet frequency domain first, and then some coefficients are modified slightly to embed the digital watermark into it.

3.2.2 A Fragile Image Watermarking Algorithm

More and more digital images appear on the web pages and portal storages. The copyright piracy and image abuses become a serious problem. In order to protect the copyright or authenticate the credibility, digital watermarking techniques grow very fast. The digital watermarking can be applied on audio, image, or video. The watermarking algorithms can be generally categorized into two kinds: robust watermarking and fragile watermarking.

The goal of robust watermarking algorithms [3.8- 3.10] is to remain the embedded digital watermarks even after some attacks. The attacks include the malicious modifications, i.e., averaging and watermark removal/counterfeit, and unintended processing, i.e., resizing, compression, and filtering. Robust watermarking algorithms are developed on the purpose of copyright protection and ownership identification.

Fragile watermarking algorithms are very sensitive to the image processing. The broken digital watermarks indicate the alteration of the suspected image. Fragile

watermarking can operate in spatial domain or frequency domain. The spatial domain fragile watermarking algorithms [3.11, 3.12 - 15] modulate the least significant bit of the image pixels to embed a watermark bit. And the spatial domain watermarking algorithms are sensitive to the image compression and processing.

DCT, DFT and DWT are the most popular transforms where the frequency domain fragile watermarking algorithms work. In frequency domain, these transforms can separate the coefficients of digital into different priorities with accordance to human perception systems. I. Cox in his paper [2.9] pointed out that the watermarks should be added on the lowest frequency coefficients in order to ensure the enough robustness. His idea is that image processing operations dare not to distort the lowest frequency severely. Some other researchers do not agree with his thought and they claimed that the distortion introduced to the low frequency domain may degrade the quality of the image severely. The basic idea of some algorithms [3.11, 3.16, 3.17] is to only watermark some of the mid-frequency coefficients and leave the most significant ones unmodified. This introduces only minor distortion to the original images and compromising the high visual quality of the images.

In this work, we take Cox's idea. For fragile watermarking algorithms, the slight alteration does not degrade the quality of image dramatically. To assure robustness, the watermark bit sequence is etched into the low-frequency coefficients of the image in wavelet domain, as illustrated in Figure. 3.2.

Quantization [3.20] methods have been deployed in watermarking algorithms [3.21] for a long time. The basic idea is to quantize a sample, k, and use the distance between the new value with the old value to reveal the embedded message. The sample, k, is firstly quantized to a predefined value, and then k is added or subtracted by d/4, according to the value of watermark, to form a new value, here d is the quantization step.

To illustrate it clearly, a process is outlined as follows:

```
Quantization (k, m)
{
        if   m ≡ 0
                k' ← q(k, d)+d/4
        else
                k' ← q(k, d)-d/4
}
```

where d is a predefined quantization step. The sample is quantized as follows:

$q(k, d) = floor(k/d)* d$

After q(*), k is quantized to a predefined value, and then k is added or subtracted to form a new value. The operation of addition or subtraction depends on the value of the watermark bit.

The detection process works as follows:

```
Detection (k')
{
        if 0< k'-q(k, d) < d/4
                m ← 0
        else if -d/4 < k'-q(k, d) < 0
                m ← 1
}
```

The watermarking algorithm works in the low frequency wavelet coefficients of an image. The discrete wavelet transform is commonly used in image processing, including digital watermarking. Embedding digital watermark in the low frequency is also commonly deployed.

Xie's Algorithm

The watermark embedding and extraction algorithm is adapted from Xie's work [3.2]. The original goal of her algorithm is for authentication instead of watermarking. The watermark embedding process of Xie's algorithm works as follows:

1) An image is transformed into a wavelet tree presentation
2) In the lowest frequency domain, a window runs from top to bottom, and line by line. The running window selects a triple pair each time
3) Quantize the middle element of the triple pair
4) Restore the wavelet tree to the space domain

The novelty of Xie's algorithm is that the quantization step is set to d/20 (step 3), where d is the distance between the largest element with the smallest element. By such excellent design, the distortion introduced by watermarking is so small that the watermarking process can be performed in the lowest frequency domain. The extraction process is similar to embedding except for substituting step 3 with Detection function as mentioned above.

Improvement

Since a fingerprint or an iris image is sparse image and many small blocks of the fingerprint image is blank. Quantizing a triple pair, whose values are nearly same, is nonsense. So in this work, the triple pair is screened, and the triple pairs having nearly same values are skipped. Xie's algorithm treats each pixel equally, thus can not be used for the purpose of watermarking. The modified algorithm avoids embedding watermark into the unstable regions.

The screen process works as follows:

1) Sort the triple pair
2) d ← largest – smallest
3) if d < largest/10

   jump to the next triple pair

   else Embedding

The modification of the embedding process can increase performance of the detection process dramatically.

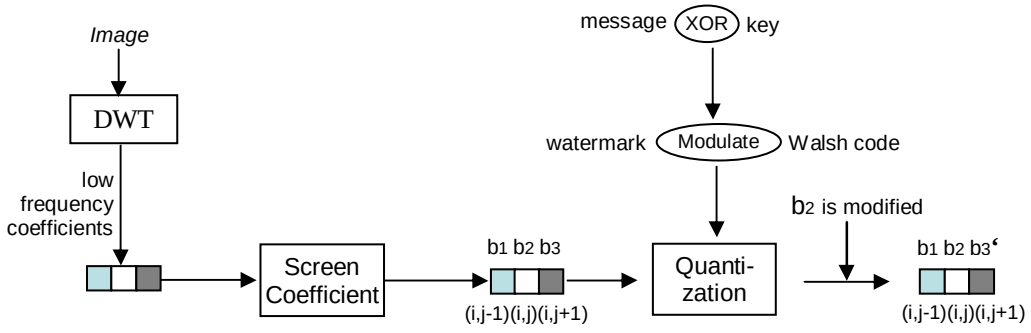


Figure 3.2: The Watermark Engraving Structure

The message containing timestamp and the secret key for this transaction are performed an XOR operation to generate the digital watermark. To further improve the performance of the watermarking algorithm, the watermark is modulated using Walsh code. That extends the power of watermark.

The improved watermarking algorithm presented here is already robust enough for the purpose of copyright protection. Some natural images have been tested to prove this.

3.3 Watermarking Algorithm

3.3.1 Principles of Progressive Watermarking

Digital watermarking is a generic method to stamp images in such a way that the original image may clearly be identified from some information hided into the data. Of course, this additional information should not be removable by some attacker. When applying this technique to identification applications in a distributed environment, the sender of the image has to insert watermarks in such a way that its unique receiver may easily decide whether the image originates from this specific sender. In addition, the correct watermark guarantees the secure transmission of the information via public networks. The general model for watermarking technology can thus be represented as

$$I' = I + W \tag{3.4}$$

Here, I' is the watermarked image, I is the original host image, and denotes W as the digital watermark inserted to I.

In this scheme, the complete image is first watermarked by the front-end device. It is then transmitted to the receiver, e.g., a server at the headquarters. The server subsequently extracts the embedded digital watermark to authenticate the received image.

Several objectives should be addressed when developing watermarking technology for low-end devices communication over public networks: 1) the watermarking procedure should be of a light-weighted computation; 2) the energy of the additive watermarks should be low enough to maintain the visual quality of the images.

The proposed approach to a secure transmission of images takes these objectives into account. It thus ensures a secure transmission, it requires a very low computing power of the front-end device, and it exploits public authentication techniques.

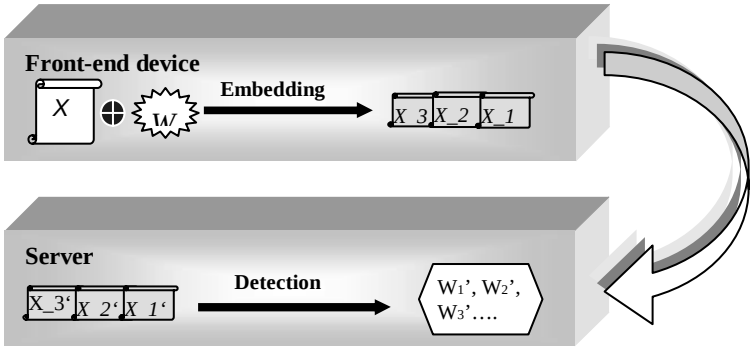


Figure 3.3: Watermarking System Based on the Progressive Transmission

We subdivide the original image I into a sequence of images clips { $I_0, I_1, I_2, \ldots, I_M$ }. The total watermark W can also be structured into a sequence of sub-watermarks, { $W_0, W_1, W_2, \ldots, W_M$ }. Each sub-watermark is then to be inserted into the corresponding image clip in order to create a watermarked clip. Thus, (3.4) can be rewritten as

$$I_k' = I_k + W_k, \qquad 0 \leqslant k \leqslant \mathrm{M} \qquad (3.5)$$

This means that a sequence of sub-watermarks can be embedded into a series of image clips. Then the watermarked image clips are sequentially transmitted to the server, which is in charge of image reconstruction and of digital watermark authentication. Finally, the watermark detector of the server extracts a sequence of sub-watermarks denoted as { W_0', W_1', W_2',…, W_M' } from the received set of marked image clips and decides on the authenticity of the transmitted image. This progressive watermarking method is depicted in Figure 3.3.

3.3.2 Architecture of Progressive Watermarking

The progressive watermarking approach outlined above has to consider the highly asymmetric properties of the envisaged distributed authentication system architecture in terms of computing power and memory space. We therefore assign all the heavy computational tasks of watermark generation and authentication checks to the back-end device, i.e., the server. This results in a transaction based, recurrent communication scheme as outlined in Figure 3.4. We focus on the watermark generation, insertion, and validation first.

WM_Server and *WM_Sender* are introduced for this purpose. *WM_Server* is a software module running on a powerful computer acting as the server of the distributed client-server architecture, and it has a RSA facility. This server is in charge of the watermark generation and of security issues. *WM_Sender* is a software module running on the front-end device acting as the client. This device consists in general of an embedded system featuring a low to medium performance micro-controller or of a low-end configurable system-on-chip such as Atmel AT94K FPSLIC family, respectively, and of an image generator. In case of an ATM banking application scenario, the front-end device produces the customer's fingerprint image from sensor data or her or his eye iris image by means of a camera, then it inserts watermarks to the scanned image and transmits the enhanced image to the central server according to the sequential procedure outlined in Figure 3.4.

3.3.3 Trustable End Device

A trustable *WM_Sender* is crucial to the security and robustness of the whole system. In order to ensure the trustability of *WM_Sender*, the *WM_Server* generates a message *M* based on pseudo-noise during the initialisation of the system, and signs this message as to get a unique signature which, in turn, is assigned to *TransSig*. The RSA private key of *WM_Server* is exploited for this purpose. *WM_Sender* then uses *TransSig* as its stream cipher's key for the entire transaction.

3.3.4 Transaction

A transaction denotes the procedure of image collection, of watermarking, of image transmission, and of image authentication. It consists of a sequence of interactions between *WM_Sender* and *WM_Server*. Each transaction has a unique signature denoted as *TransSig*. A cycle defines the manipulation and the subsequent transmission of one image clip only. Thus, a transaction embodies several cycles depending on the size of both the entire image and the size of a single clip. The interaction of *WM_Sender* and *WM_Server* during a transaction is detailed in Figure. 3.4.

At the beginning of the transaction, *WM_Server* creates a message sequence { M_0, M_1, M_2,……, M_m} based on random pseudo-noise data, and signs this message M by encrypting it with the RSA private key. Upon completion, *WM_Server* sends the encrypted message sequence denoted as { eM_0, eM_1, eM_2,…, eM_m } to *WM_Sender* one by one during the transaction.

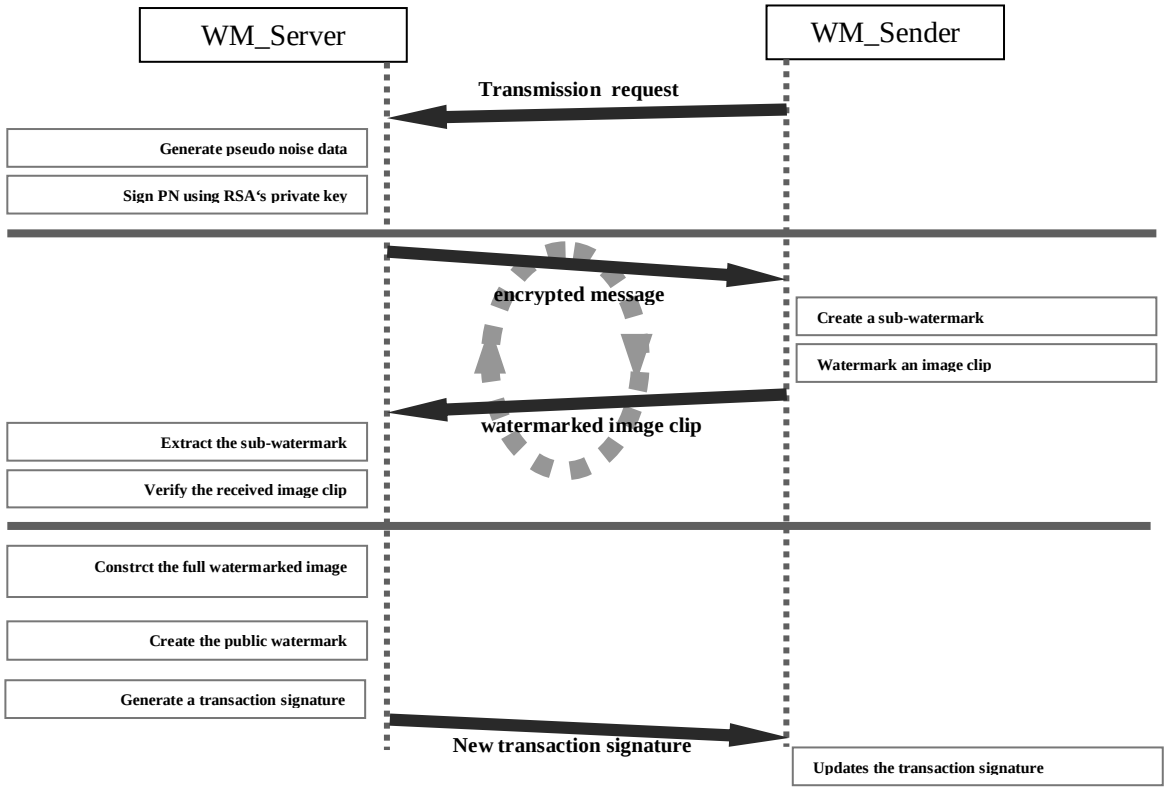


Figure 3.4: Outline of a Transaction

At *WM_Sender*'s side, one piece of message, eM_i, is received and then signed by *TransSig*. Later on, *WM_Sender* generates a series of digital sub-watermarks. The image is first subdivided into several clips. Then, the watermark embedder on *WM_Sender* inserts one sub-watermark into the associated image clip. The watermarked image clip is sent to the *WM_Server* immediately. *WM_Sender* repeats all these operations until all image clips have been sent out.

After accepting all watermarked image clips, *WM_Server* reconstructs the full image and checks the authenticity of the received image as follows.

3.3.5 Generation of Digital Watermark

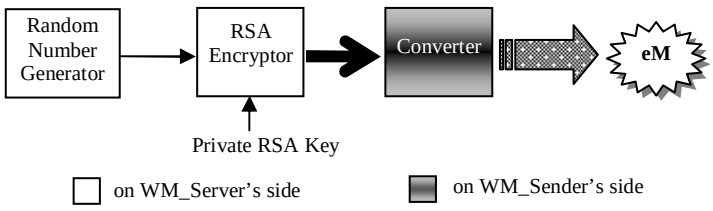


Figure 3.5: Digital Watermark Generation

The watermark generator within *WM_Server* produces pseudo-noise data consisting of 100 key words. This key sequence is then signed by applying RSA encryption with the

private key of *WM_Server*. This module forwards the encrypted message sequence *eM* to *WM_Sender* as outlined in Figure 3.5.

Once *eM* is received by *WM_Sender* then its stream cipher signs it using *TransSig* in order to create the final digital sub-watermark which, in turn, is inserted by *WM_Embedder* into the coefficients sequence of the image clip at hand as depicted in Figure 3.6.

3.3.6 Authentication

The authentication of an image is addressed in two ways by the proposed approach to progressive watermarking – real-time authentication and post authentication.

3.3.6.1 Real-time Authentication

In order to ensure the overall safety of the communication, real-time authentication is employed during the transmission. In the beginning of each cycle, *WM_Sender* generates a digital watermark, w_i, using stream cipher, *TransSig* as the key. w_i is inserted into the i-th clip of the original image, and then the watermarked clip is sent to *WM_Server*.

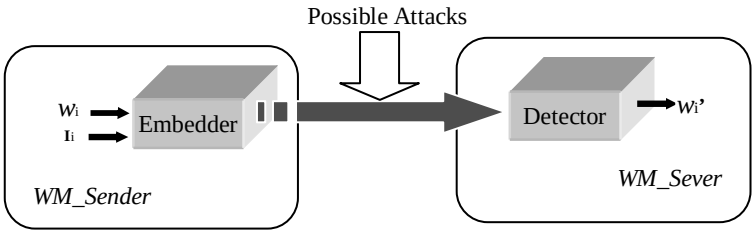


Figure 3.6: Real-Time Authentication

As soon as *WM_Server* receives the watermarked image clip, it extracts the embedded watermark, w_i', immediately. Since *WM_Server* holds the signature for each transaction, *TransSig*, it can sign the encrypted message sequence *eM* using it, to re-construct the same sub-watermark *WM_Sender* embedded previously in the image clip. *WM_Server* then compares the complete extracted watermark *W'* with the original one, *w*. If *w'* does not differ from *w*, then the received image clips are composed into the authentified final image.

Finally, *WM_Server* generates the *public_WM* by selecting a part of the full watermark sequences. Later on, *public_WM* is published.

3.3.6.2 Post Authentication

Post authentication as illustrated in Figure 3.7 is intended to be used by anyone. A suspected image can be authenticated by assessing the similarity between the embedded watermark and *public_WM*.

Since *public_WM* has initially been signed by means of the RSA private key of *WM_Server*, the validation of *public_WM* can be authenticated too by decrypting *public_WM* using *TransSig* (by stream cipher) and by applying a RSA encryption operation exploiting WM_*Server*'s RSA public key. Post authentication is performed by some auxiliary tools denoted as *Image_Authenticator* and *WM_Validator*, respectively. *Image_Authenticator* extracts the digital watermark from the watermarked image and then compares it to the public key *public_WM* in order to make an assessment of authenticity. In contrast, *WM_Validator* is used to authenticate the author of *public_WM*. Everyone can use these tools in order to authenticate suspected images and *public_WM*s. Finally, one characteristic property of the proposed approach should be mentioned - both *public_WM* and *WM_Server*'s public RSA key are accessible to anybody. So, both image and watermark authentications are public.

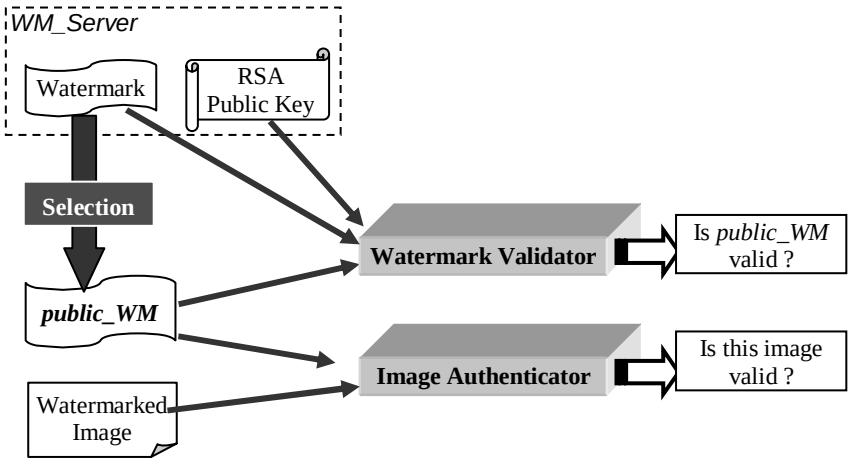


Figure 3.7: Post Authentication

3.4. Analysis of Workload

The workload on *WM_Server* consists of management and detection tasks of digital watermarks. In contrast, the workload on *WM_Sender* deals with watermark management and embedding. When looking more into the details of watermark management on the *WM_Server* side, the tasks of watermark generation, of RSA encryption, and of stream cipher

encryption have to be executed. The watermark management on *WM_Sender* side consists of stream cipher encryption.

In our scheme, the *watermark embedder* of *WM_Sender* takes the same computation load as the *watermark detector* of *WM_Server* does. The difference of workload, τ, between the workloads of *WM_Sender* and of *WM_Server* reads as follows

$$\begin{aligned}\tau &= \tau_{server} - \tau_{sender} \\ &= (\tau_{svr\_wm} + \tau_{detection}) - (\tau_{sender\_wm} + \tau_{embedding}) \\ &= \tau_{svr\_wm} - \tau_{sender\_wm} \\ &= (\tau_{PN\text{-}generation} + \tau_{rsa} + \tau_{wm\_gen}) - (\tau_{wm\text{-}gen}) \\ &= \tau_{PN\text{-}generation} + \tau_{rsa} \end{aligned} \quad (3.6)$$

$$\text{Therefore,} \quad \tau \approx \tau_{rsa} >> \tau_{sender} \quad (3.7)$$

From (3.7) it is obvious that the workload on *WM_Server* is far heavier than that on *WM_Sender*. The proposed progressive watermarking approach considers the in general highly asymmetric properties of distributed authentication architectures in an appropriate way.

3.5 Experimental Results

The outlined approach to progressive watermarking has been implemented as a prototype. It consists of three software packages, namely *WM_Sender*, *WM_Server*, and *Post_Authentication_Tool* as outlined in Figure 3.8:

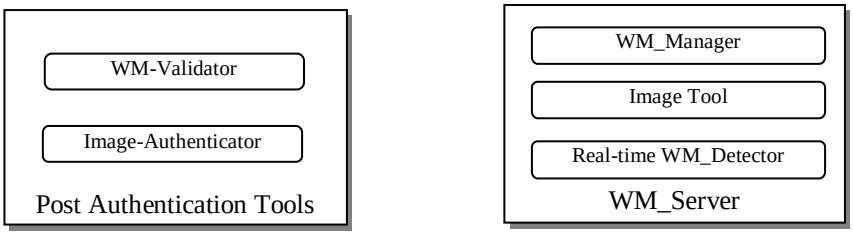


Figure 3.8: Software Components of Progressive Watermarking

- ***WM_Sender*** decomposes the original image into clips, inserts digital watermarks into their DWT coefficients, and then sends them to *WM_Server*.
- ***WM_Server*** generates digital watermarks, receives the watermarked images clips, and extracts the embedded marks. In addition, it is responsible for the recording of the complete digital watermark and the entire image,

- ***Post Authentication Tools*** can be used by everyone to verify the validity of *public-WM*s and to authenticate suspected images.

Several publicly available images were used to test the performance of this system.

3.5.1 Visual Effects of Watermarked Images

(a) Watermarked Fingerprint

(b) Sharpened Difference between watermarked one with the original

(c) Watermarked 'Lena'

Figure 3.9: High-Qualified Watermarked Images

Figure 3.9 shows the high-qualified watermarked images, and it is difficult to find the visual difference between original images and watermarked images by inspection only.

3.5.2 Post Authentication

The verification of the suspected image is performed by using *Image_Authenticator*. It first extracts the suspected watermark from the suspected image, and computes the correlation as follow:

$$\text{Correlation} = \frac{\sum_{j=1}^{M-1} (W_j \times W_j')}{\sqrt{\sum_{i=0}^{M-1} (W_j \times W_j) \sum_{i=0}^{M-1} (W_j' \times W_j')}} \qquad (3.8)$$

Table 3.1: Correlation

	Correlation (image without the digital watermark)	**Correlation (watermarked image)**	**PSNR**
Fingerprint	0.363892	0.865455	41.448132
Lena	0.402415	0.929432	42.124886

PSNR: Peak Signal to Noise Rate

From the table above, we know *Image_Authenticator* can distinguish the watermarked images from images without the correct digital watermarks. The *WM_Validator* detects any changes on the public watermarks, even a very slight modification.

3.5.3 Overhead Introduced by Proposed Scheme

Table 3.2: Overhead Introduced by Proposed Scheme

	WM_Server	**WM_Server (without RSA operation)**	**WM_Sender**
Time in second	161.34	10.21	10.23
Overhead in second	151.21	0.1039	0.1053
Overhead/Time	93.72%	1.02%	1.03%

All results from our prototype run on Pentium 3 700 MHz processor under Linux

In our experiment, a pair of 43-byte-length RSA keys was used. The table above shows the overhead introduced by our scheme. It is clear that *WM_Server* spends quite a lot of computation time on digital watermark generation and the workload of *WM_Server* is far heavier than that of *WM_Sender*.

3.6 Conclusion

This proposed scheme has several advantages and is suitable for real-time image collection: The architecture of the proposed scheme is asymmetric, *WM_Server*'s computation task is heavy, while *WM_Sender*'s computation task is rather light. In addition, the image authentication is public, and a verification of public watermarks is also public. System-on-chip technology provides a very convenient platform for low-end embedded devices. We are working towards an implementation of the client as an embedded system by means of a configurable SoC device.

Chapter 4 Authentication Methods for Speech

There are two kinds of security mechanisms, feature-based digital signature and speech watermarking, can be exploited for the purpose of integrity and confidentiality. Feature-based digital signature, which is similar to the traditional hash-based digital signature, relies on a checksum outside of the data. Speech watermarking embeds digital watermarks into the speech. The speech signature was the first method developed for speech authentication in my work and then a watermarking algorithm with use of frequency domain linear prediction and a watermarking algorithm based on deterministic and stochastic decomposition were exploited. But both of them are not the very convenient solutions for market-available speech communication system, thus a speech watermarking algorithm incorporating with GSM 610 coder is presented at the last part of this chapter.

Speech production [4.20] and linear prediction [4.20] is the supporting techniques for speech watermarking. So a short introduction about linear prediction is necessary here. It is clear that human's lung pushes the air through the vocal tract and mouth to generate speech. The position of vocal tract and oral cavity change the speech. Human's vocal tract, nasal cavity and oral cavity can be modelled as a linear filter, and the lung can be viewed as a pump. A speech production model is shown in Fig. 4.1. To construct the linear predictor, a series linear parameters need to be extracted.

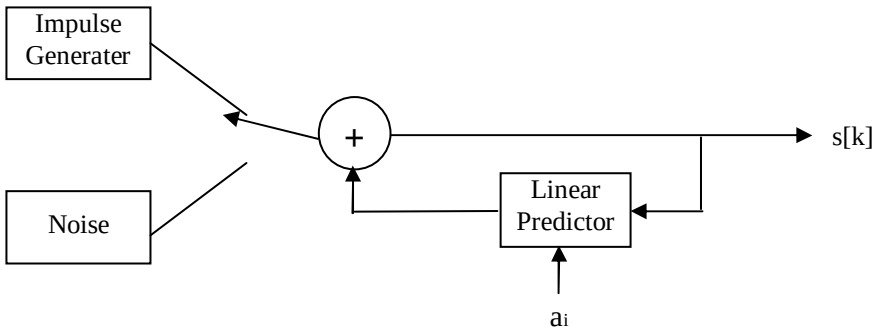


Figure 4.1: Linear Predictive Coding

Linear prediction is deployed in speech analysis and coding for more than 30 years. It assumes the current speech frame can be estimated based on the n-th previous speech frames, and the formula is:

$$s[k] = \sum_{i=1}^{N} a_i * s(k-i) + Gw(k) \qquad (4.1)$$

$\{a_1, a_2, a_3, \ldots, a_N\}$ are the parameters of the linear filter. The linear filter determines the formant of a speech frame thus that the parameters are the salient features of a speech frame.

4.1 A Digital Signature Scheme for Speech

Creating a digital signature is the most straightforward way to authenticate the integrity and confidentiality of speech. Some modification should be under consideration when implementing hashing-based schemes for speech. First, since speech is a series consisting of multiple frames, a subtraction of the middle of a speech may damage the semantic meaning while a truncation of the unvoiced tail is acceptable. A hashing-based authentication method is illustrated in the following.

There are several spectral parameter sets, i.e., LPC parameters, log area ratios, Mel scale cepstrum coefficients, and power density spectrum coefficients, can be used to depict the features of a speech. Among them, LPC parameters [4.15] have been used for a long time. LPC parameters model the structure of vocal organs, therefore they are be widely used to describe the features of a speech.

Linear predictive coding is commonly used by voice encoders and decoders. Since the LPC coefficients are sensitive to distortion, some other advanced representations of linear prediction are proposed. The most commonly used advanced representations include log area ratios (LAR), line spectral pairs (LSP), linear spectral frequency (LSF) and reflection coefficients (RC). Among the multiple representations, LSP and LSF have been widely used due to its stability of the predictor. LSF parameters [4.20] can be computed from LPC coefficients directly.

As we mentioned in the early part of this thesis, message authentication code does not have any flexibility to the minor errors, so that a more flexible authentication method for multimedia data is needed.

The proposed speech authentication algorithm is illustrated in Fig. 4.1. A whole speech is split to multiple blocks and each block lasts for 20ms. Then the parameters of the filter for a block are estimated. A bulk comprising the parameters of 50 filters is applied with forward SVD transform.

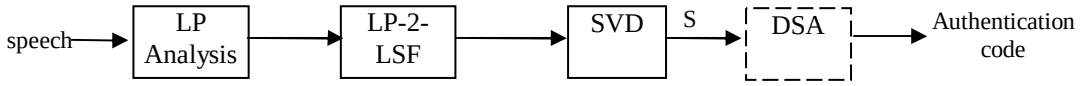


Figure 4.2: Watermark Embedding Algorithm

Mathematically, a matrix, X, can be decomposed as X = U*S*V', which is so-called Singular Value Decomposition [4.21]. The matrix S is singular, thus the singular vector of S can be used to represent the matrix uniquely. SVD is popularly used since its high resolution properties and its insensitivity to model order overestimation.

At last, an authentication code can be generated based on the singular vector. Then the authentication code is sent to the receiver complying with the corresponding speech bulk. The SVD process of LSF loose the sensitiveness of the LPC filters thus the proposed scheme is more flexible than message authentication code.

The receiver does the linear predictive coding on the corresponding speech block and then decomposes the LSF parameter matrix using SVD to generate the singular vector, S'. Measure the similarity between the received singular vector, S, with the extracted singular vector, S'. Denote S as s[i] and S' as s'[i], $0 \leq i < N$. The similarity is calculated as follows:

$$sim[k] = 1 - \frac{1}{N} \sum_{i=0}^{N-1} (s[i] - s'[i]) / \ s[i] \qquad (4.2)$$

A prototype has been implemented using Matlab. Since hashing-based methods are not our interest, the further analysis did not perform on this scheme.

4.2 A Speech Watermarking Algorithm by Frequency Domain Linear Prediction

4.2.1 Commonly-Used Skills for Audio Watermarking

A variety of approaches already exist to embed information into audio data. The techniques range from the simple LSB method to the spread-spectrum methods. In this section, a speech watermarking algorithm using linear prediction coding is presented.

Since the human auditory system is higher sensitive than human visual system, the watermarking algorithms for audio data have limited categories. The audio watermarking algorithms can be classified into two general kinds:

1) Utilizes the weakness of human auditory system. LSB, echo hiding and phone coding LSB (short for Least Significant Bit) coding [4.2, 4.3, 4.4, 4.5] take the advantage of precision limitation of human auditory system; LSB has a low robustness while the work load of watermarking algorithm is rather low. The phase coding method was presented in [4.1, 4.6, 4.7]. The basic idea is that human are not sensitive to the change of phone. Echo hiding [4.8, 4.9] utilizes the temporal post masking of human auditory system.

2) By digital signal processing skill. I. Cox is the first deployed spread spectrum technique in digital watermarking [4.10]. The advantage of spread spectrum watermarking is that the distortion is distributed over a great number of parameters while the modification on each value is very small. Later, spread spectrum technique was widely used in audio, video, image watermarking algorithms [4.10, 4.11]. Sinusoidal modulation [4.22] utilizes the orthogonality between sine wave with different frequency.

4.2.2 Linear Prediction in Frequency Domain

Linear prediction usually it works in time domain thus that it was called “time domain linear prediction (TDLP)”. In 1996, J. Herre introduced linear prediction method in frequency domain (FDLP) [4.16]. In his original work, a signal is firstly converted to DCT coefficients and then is applied linear prediction. Later, in 2002, FDLP is included in AVS audio-video coding standard [4.18]. M. Athineos and D. Ellis at Columbia University used FDLP to extract the sound texture [4.17]. Mathematically, FDLP can be formulated as follows:

$$S' = \mathrm{LP}(\ \mathrm{DCT}\ (\ s\ )\ ) \tag{4.3}$$

FDLP works as follow:

1) apply DCT transform on a signal sequence
2) divide the coefficients into multiple sub-band
3) do linear predictive coding in each sub-band

Then one can do analysis on the coefficients matrix of sub-bands or on the spectral representation. By FDLP, some new features can be developed to present audio processing (the author believes potentially it would be a powerful tool for applied signal processing). In

author's view, DCT can generate the frequency spectrum suitable for a signal, and linear predictive method working in frequency domain can describe the frequency pattern generation. Thus FDLP presents the rich temporal features of speech in frequency domain.

4.2.3 Speech Watermarking Algorithm Using FDLP

To watermark a speech, an algorithm using linear prediction in frequency domain is presented in this subsection. Discrete wavelet transform is exploited in the proposed algorithm. Discrete wavelet transform can divide the speech signal into a multiple sub-band presentation which is well suited for psychoacoustic model of human auditory system. In addition, some wavelet sub-band can still hold some useful important features of speech, such as the shapes, even in frequency domain. That makes discrete wavelet transform a better choice than the short time Fourier transform in signal processing.

In our scheme, a whole speech is split to a block sequence by running a hamming window through the speech. In wavelet domain, the detail part of a speech block, denoted as *D[k]*, describes the high frequency components, while the approximation part, *A[k]*, depicts the more smooth part of a speech. Further decompose *A[k]*, the speech can be represented as:

$$s[k] = AA[k] \oplus AD[k] \oplus DD[k] \quad (4.4)$$

The approximation part, *AD[k]*, can be viewed as a stationary signal and be transformed to a linear predictive presentation. The watermark is embedded in the second-low frequency part, *AD[k]*. Then the watermarked second-low frequency part, *AD' [k]*, is synthesized with the other frequency part to generate the watermarked speech.

$$s'[k] = AD' [k] \oplus AA [k] \oplus DD [k] \quad (4.5)$$

For a stationary signal, *s[k]*, linear prediction synthesis can reconstruct a new one, *s' [k]*, which is quite similar to the original, thus the watermarked speech, *s'[k]*, has a high fidelity.

The embedding and extraction procedure is performed on each frame and is presented as follows:

Embedding Algorithm

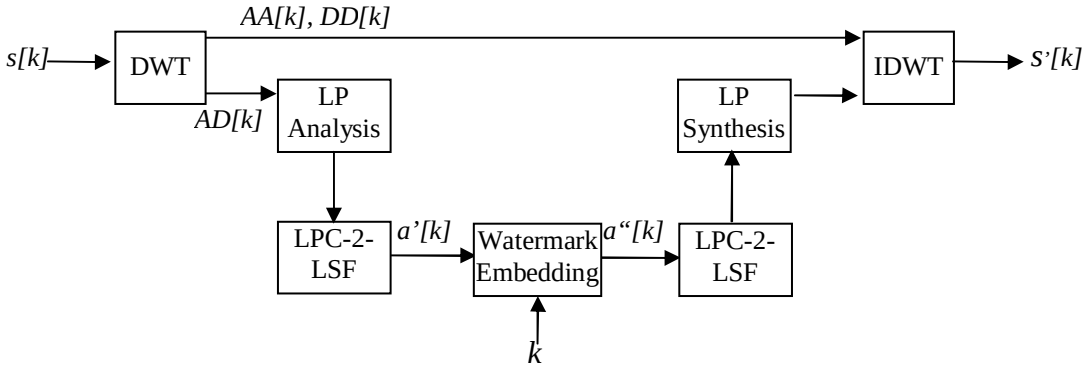


Figure 4.3: Watermark Embedding Algorithm

As illustrated in Figure 4.3, a speech block, denoted as *s[k]*, passes a DWT filter and is divided into a wavelet representation, select *AD[k]*. Then *AD[k]* is transformed into a sequence of LPC coefficients and the LPC coefficients are mapped to LSF coefficients, *a'[k]*. The LSF parameters are modified to insert the watermark, and are then mapped to LPC coefficients. LP synthesizer reconstructs the watermarked component. Reconstruct the watermarked speech block *s'[k]* by inverse DWT.

Since a slight modification on LPC coefficients can make the LP filter unstable, we transform the LPC coefficients to linear spectral frequency (LSF) parameters. LSF parameters model an all-pole filter, and have a property of local distortion which makes it more stable than LPC coefficients.

The watermark embedding is based on additive method. Firstly, a LSF parameters is selected and then a small variable, $\alpha * w$, is added to the selected candidate. In this work, α = 0.1. The watermark embedding process is described as follows:

Input: a piece of speech, *S*

a watermark bit sequence, *w[0,…,N-1]*

an index sequence, *i[0,…,N-1]*

(1) split the whole speech into *N* blocks

s[k] ← *S**Hamming_Win, where -1 < *k* < *N*

k ← 0

(2) transfer one block to wavelet domain

if k = N return end

(*A[k]*, *D[k]*) ← DWT(s[*k]*)

(*AA[k]*, *AD[k]*) ←DWT(*A[k]*)

(3) linear prediction

$a\,[k] \leftarrow \mathrm{LPC}(AD[k])$, where $-1 < k < N$

(4) convert to LSF parameters

$a'[k] \leftarrow \mathrm{LPC2LSF}(a\,[k])$

(5) watermarking

if $w \equiv 0$ $\quad a''[i[k]] \leftarrow a'[i[k]] + \alpha * w$

else $\quad a''[i[k]] \leftarrow a'[i[k]]$

(6) convert back to time space

(7) $k \leftarrow k+1$, goto step (2)

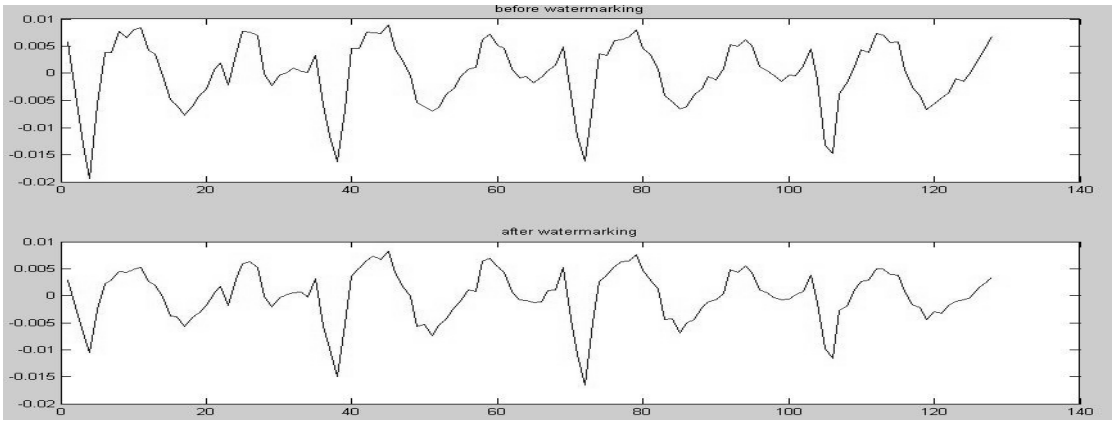


Figure 4.4: A Speech Frame: Original One (upper), Watermarked One (lower)

Figure 4.4 shows two speech blocks. The watermarking algorithm really modifies some niches of the speech.

Extraction Algorithm

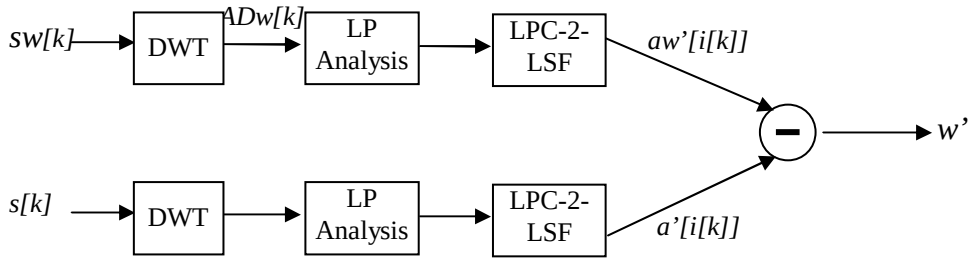


Figure 4.5: Watermark Extraction Algorithm

As illustrated in Figure 4.5, a watermarked speech block, denoted as $s'[k]$, passes a two-layer DWT filter and is divided into a wavelet representation, *AD[k]*. Then *AD[k]* is transformed into a sequence of LPC coefficients and the LPC coefficients are mapped to LSF parameters, *a'[k]*. A corresponding original speech block is performed with FDLP, and gets *a[k]*.

The watermark extraction is based on subtractive method. The extraction procedure is described as follows:

Input: a piece of watermarked speech, *Sw*

a corresponding original speech, *S*

an index sequence, *i[0,…,N-1]*

(1) split the whole speech into *N* blocks

sw[k] ← *Sw**Hamming_Win, where -1 < *k* < *N*

s[k] ← *S**Hamming_Win, where -1 < *k* < *N*

k ← 0

(2) transfer one block to wavelet domain

if k = *N* return end

(*A[k]*, *D[k]*) ← DWT(*s[k]*)

(*AA[k]*, *AD[k]*) ←DWT(*A[k]*)

(3) linear prediction

a [k] ← LPC(*AD[k]*), where -1 < *k* < *N*

aw[k] ← LPC(*ADw[k]*)

(4) convert to LSF parameters

a'[k] ← LPC2LSF(*a [k]*)

aw'[k] ← LPC2LSF(*aw [k]*)

(5) extraction

w← (*aw'[i[k]] - a[i[k]]*) / α

(6) *k* ← *k+1*, goto step (2)

A watermark bit sequence, *w*, can be extracted. To estimate the bit error rate, the formula below is used.

$$BER = 1 - count(w, \beta)/N, \quad (4.6)$$

where β is a threshold, count(*) counts the elements whose values is greater than the threshold. In this study, β = 0.8.

4.2.4 Experimental Results

In our test, the size of a frame is 256 samples, and for the 8khz test speech, one frame lasts for 32ms. Although some details are changed, the overall quality of the watermarked speech is still higher than LPC coded one. The robustness of watermarking algorithm is satisfied, see Table 4.1.

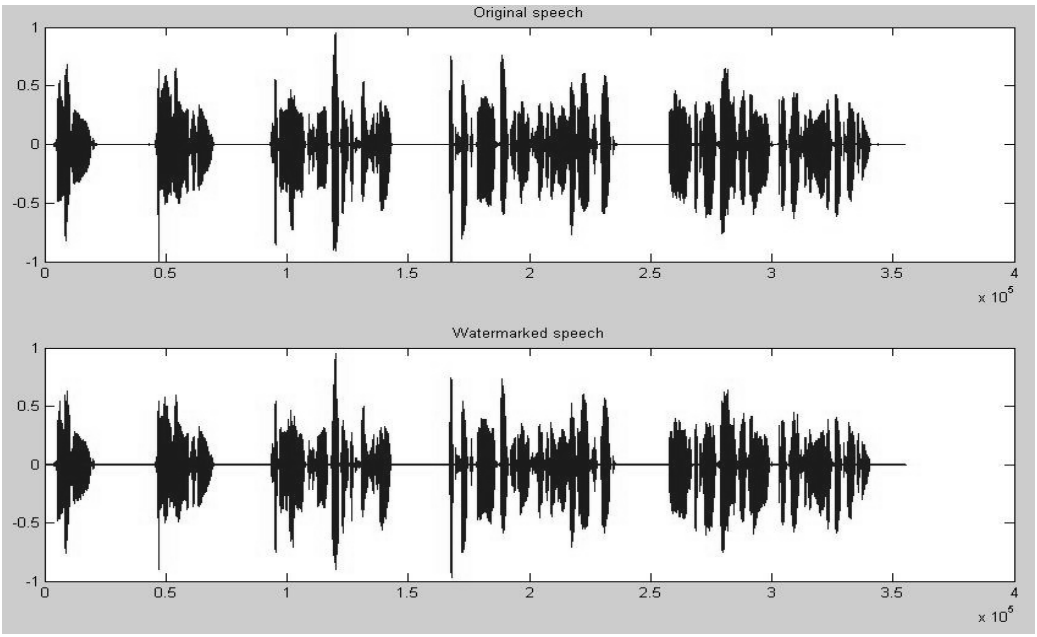


Figure 4.6: Original Speech (upper) and Watermarked Speech (lower)

Table 4.1: Bit Error Rate

	Original	**Adding 5% Gaussian noise**	**Adding 10% Gaussian noise**
"att_01.wav"	0.0370	0.0451	0.0503
"att_02.wav"	0.0216	0.0265	0.0297

In the low frequency, the frequency pattern is relative stable even having been distorted severely. FDLP presentation can describe the pattern of frequency combination, thus watermarking the LSF parameters in second-low frequency is robust to attacks. To improve

the quality of watermarked speech, the watermarking procedure should be performed on the detail part of the approximation part of level 3 wavelet composition.

To implement a blind watermarking algorithm, quantization method can be used. In addition, in order to improve the robustness of watermark detection, Walsh or Gold codes can be used to modulate the watermark. The watermarking algorithm using FDLP is still far from the integration with the commercial voice coders, so in the last sub-section, a speech watermarking algorithm is proposed which can incorporate with GSM 610 voice coder.

4.3 A Watermarking Algorithm Using Deterministic Plus Stochastic Model

4.3.1 Deterministic Plus Stochastic Model

In last sub-section, a watermarking algorithm for speech is presented. Now, we come to the watermarking algorithm for music. Nearly all sound can be presented as a deterministic part and a stochastic pat. This model, proposed by [4.24], is an improvement to the well known sinusoidal model by T. McAulay and J. Quatieri [4.23]. The deterministic plus stochastic model is formulated as:

$$S = D + E \tag{4.7}$$

Since most of the audio signals are periodic, we can use a summary of sine wave to simulate the periodic part of the audio signals.

$$d[n] = \sum_{i=1}^{N} A_i * \cos(w_k * n + \theta) \tag{4.8}$$

The different between the original audio with the sine-wave synthesized audio signal is the residual part of the signal. Usually we call it the stochastic part. The stochastic part of music includes brute of flute and the natural sounds. Figure 4.7 shows a music piece and its stochastic part.

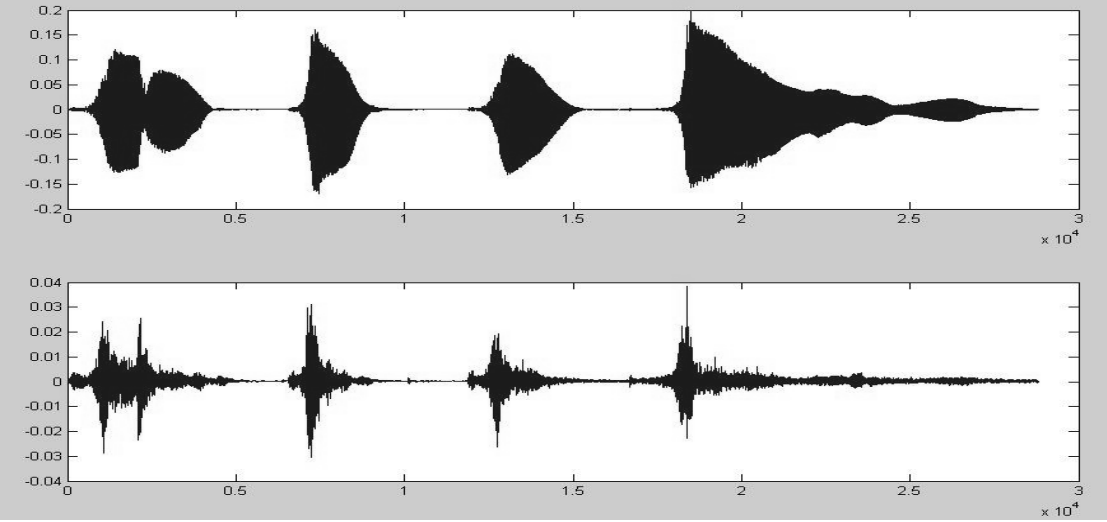


Figure 4.7: The Original Audio (upper) and the Stochastic Part (lower)

The deterministic part represents the most prominent part of an audio signal while the stochastic part is a noise like signal. The energy of the deterministic part is far heavy than the stochastic part. This gives us a hint to insert a digital watermark into the stochastic part of the audio and still keep the high quality of the audio signal. The stochastic part is also important for the high quality commercial music.

Many people from music community decompose the existing music, and apply some modifications, transforms, or enhancements on the specific parameters. Then re-synthesize the music. This is a commonly used method in music post processing. Following this method, we decompose an audio signal, modify some parameters to engrave the digital watermark, and lastly re-synthesize the audio.

4.3.2 Audio Watermarking on Stochastic Part

Watermark Embedding Algorithm

To watermark the stochastic part of an audio signal, the original audio signal should be separated to a deterministic part and a stochastic part. Then insert the watermark into the not important part of the stochastic part.

The embedding process can be illustrated as follows:

1) separate the audio signal to a deterministic part and a stochastic part
2) convert a block into wavelet domain
3) insert the watermark into the low frequency part of the wavelet coefficients

4) convert the watermarked wavelet coefficients back to time domain
5) add the watermarked stochastic part with the deterministic part to form the watermarked audio signal

The embedding process can be illustrated as Figure 4.8.

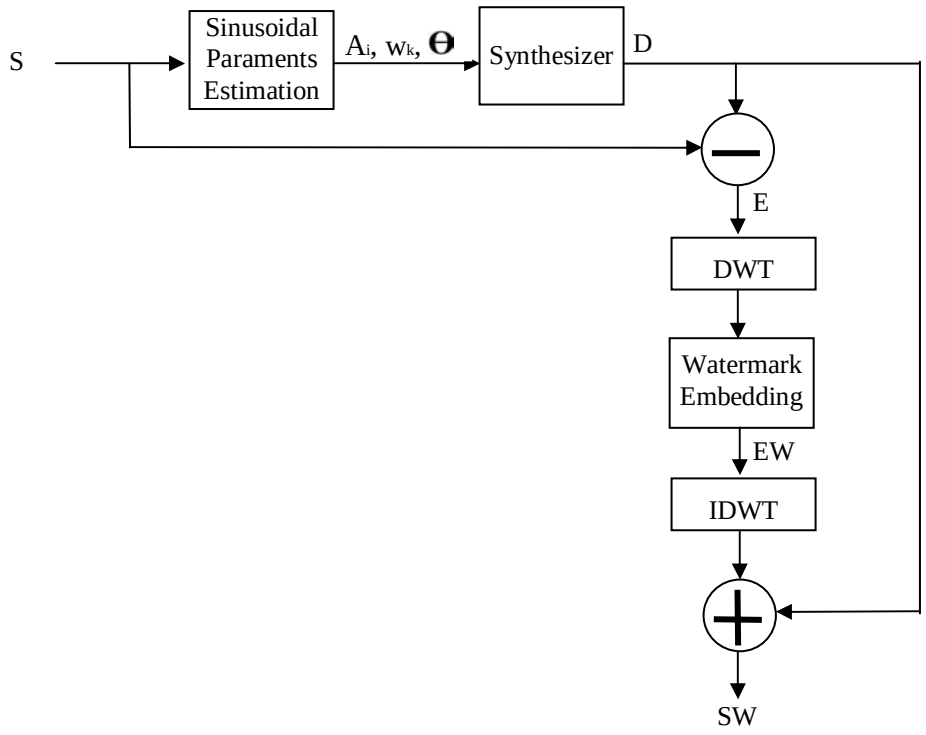


Figure 4.8: The Watermark Embedding Algorithm

The first step, deterministic and stochastic decomposition, applies the short time Fourier transforms on the audio signal. Then search the frequency tracks in the spectral representations to detect the amplitudes, phases and frequencies of the sine waves. The details of this decomposition can be seen in [4.24]. Then feed the parameters (amplitudes, phases and frequencies) to a sine oscillator to synthesize the deterministic part, d[n]. Lastly, subtract the original audio signal with the synthesized audio signal to retrieve the stochastic part, e[n]. Part of the Matlab codes for this step is adapted from the source code from Speech and Audio Processing Group of Columbia University.

The stochastic part can be extracted either in frequency domain or in time domain. In this work, the stochastic part is extracted in time domain. Figure 4.9 shows the spectral distribution of the original audio signal with its stochastic part. The prominent part has been

eliminated from the stochastic part. Although the stochastic part is noise like, it really plays an important role for the high quality audio products.

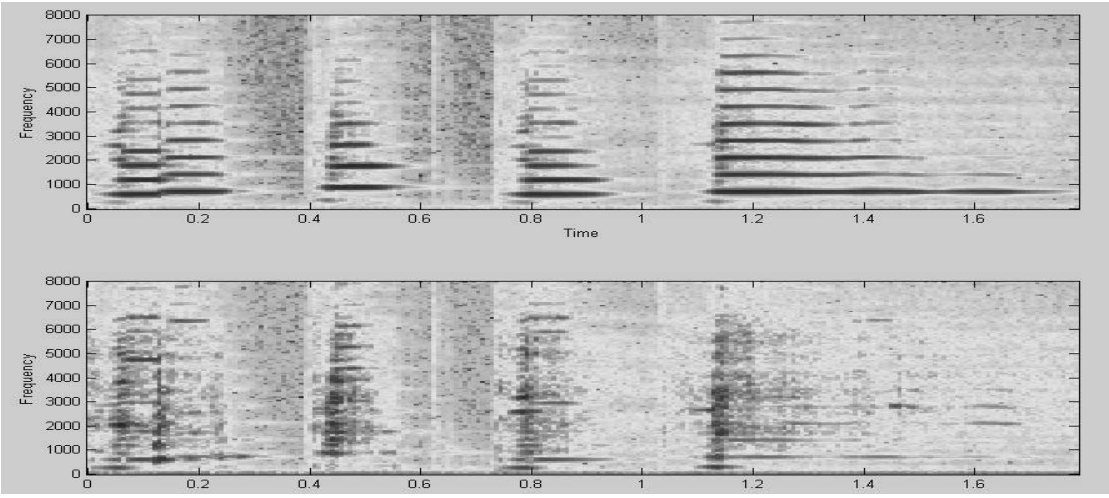


Figure 4.9: The Original Audio (upper) and the Stochastic Part (lower)

As mentioned earlier in this thesis, the wavelet transforms can separate the signal into multiple sub-bands. The distortion introduced in the middle frequency coefficients does not degrade the quality of audio signal and not easily be eliminated. To embed the digital watermark, w, into the stochastic part, e[n], the wavelet transform is employed first and then modify the middle frequency coefficients. Lastly, converts the watermarked coefficients back into the time domain. We get the watermarked stochastic part till now. The details are depicted as follows:

Input: the stochastic part of the audio signal, *E*

a watermark bit sequence, *w[0,…,N-1]*

(1) split *E* into *N* blocks

e[k] ← *S**Hamming_Win, where -1 < *k* < *N*

k ← 0

(2) transform one block to wavelet domain

if k = N return end

(*A[k]*, *D[k]*) ← DWT(*e[k]*)

(*DA[k]*, *DD[k]*) ←DWT(*D[k]*)

(3) watermarking

if *w[k]* ≡ 0 *DD[k]* ← *0*

else $DD[k] \leftarrow DD[k]$

(4) convert back to time space

(5) $k \leftarrow k+1$, goto step (2)

Figure 4.10 show the original stochastic part and the watermarked one. They are quite similar. Since the stochastic part is noise like, so the modification on the low frequency does not degrade the stochastic part severely.

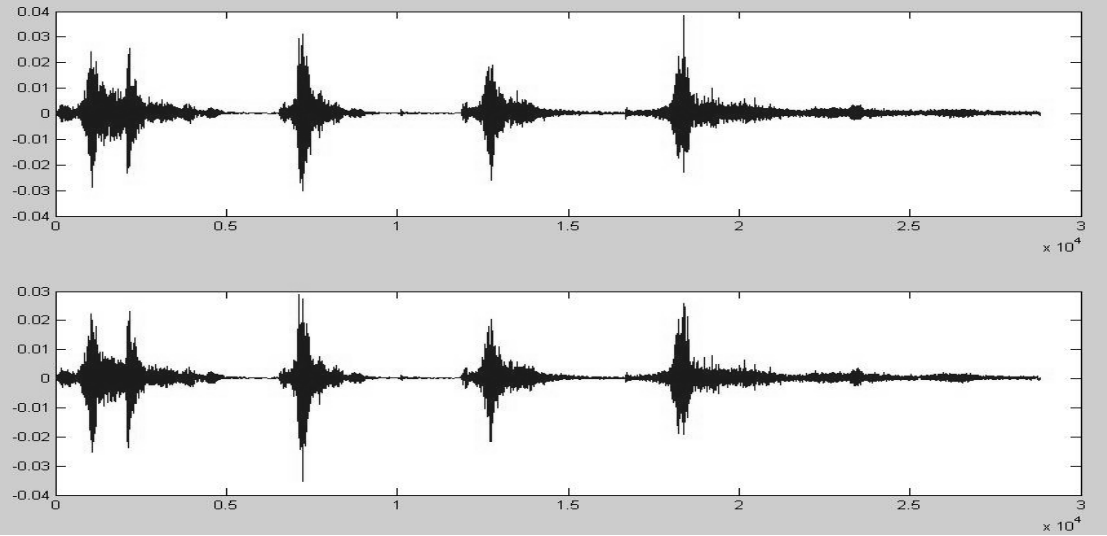


Figure 4.10: The Original Stochastic Part (upper) and the Watermarked Stochastic Part (lower)

Now, add the watermarked stochastic part, EW, with the synthesized deterministic part, D, to form the watermarked audio signal, SW. Figure 4.11 shows the original audio signal and the watermarked audio signal. The watermarked audio signal is quite similar to the original audio signal.

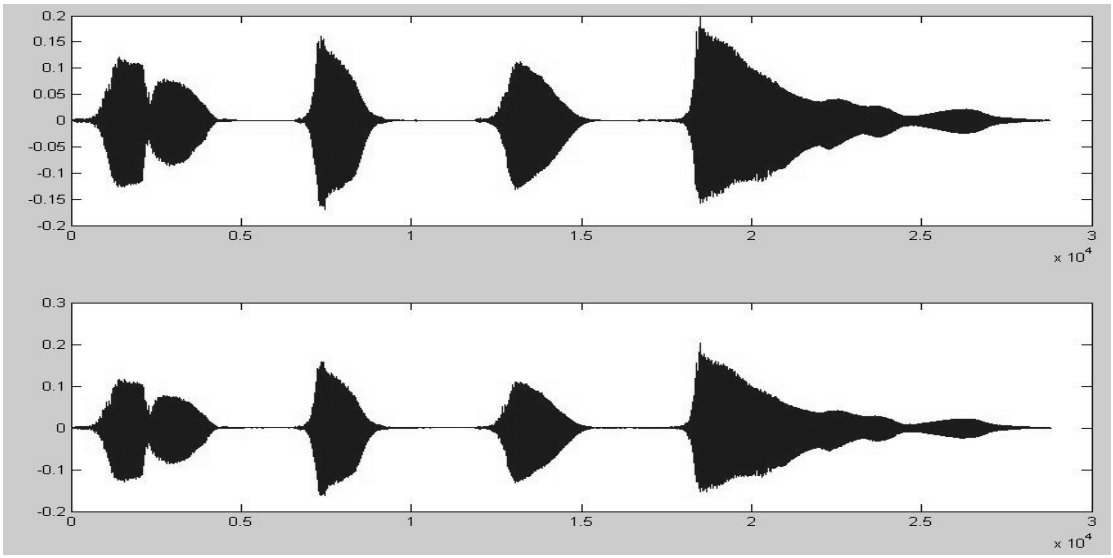


Figure 4.11: The Original Audio (upper) and the Watermarked Audio (lower)

Watermark Extraction Algorithm

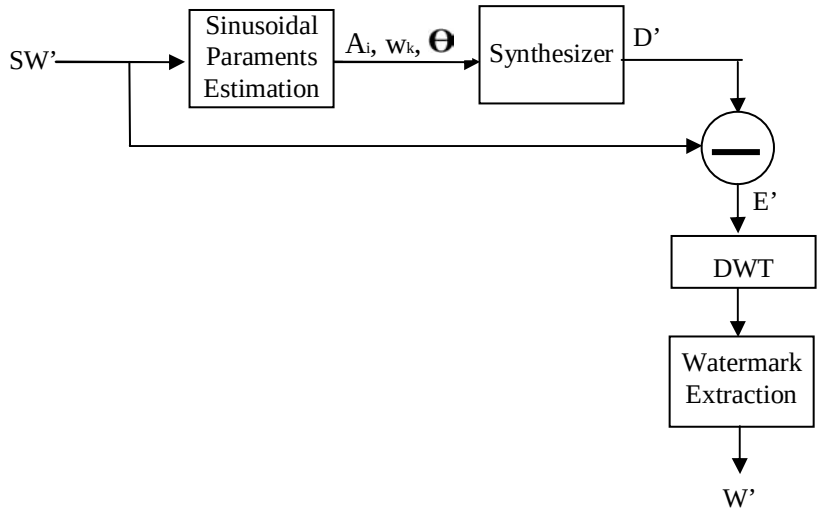


Figure 4.12: The Watermark Embedding Algorithm

As illustrated in Figure 4.12, a watermarked audio signal, denoted as *SW*, passes a sinusoidal parameters estimator to detect the amplitudes, the phases, and the frequencies. Then synthesize the deterministic part, D'. Subtract the deterministic part from the original watermarked audio signal to get the watermarked stochastic part, E'.

The watermarked stochastic part, E', is fed to a two-layer DWT filter and is divided into a wavelet representation. Extract the embedded digital watermark, W', from the low frequency coefficients. The extraction procedure is described as follows:

Input: the watermarked stochastic part, *E'*

Output: the extracted watermark, *w*[0..N-1]

(1) Split *E* into *N* blocks

e[k] ← *E'**Hamming_Win, where -1 < *k* < *N*

k ← 0

(2) Transfer one block to wavelet domain

if k = *N* return end

(*A[k]*, *D[k]*) ← DWT(*e[k]*)

(*DA[k]*, *DD[k]*) ←DWT(*D[k]*)

(3) Extract the embedded watermark

if mean(*DD[k]*) < α *w[k]* ← 0

else *w[k]* ← 1

in this study, α = 0.01

(4) *k* ← *k+1*, goto step (2)

A watermark bit sequence, *w*, can be extracted. To estimate the error rate, the formula below is used.

$$ER = count(w)/N, \quad (4.9)$$

4.3.3 Experimental Results

Figure 4.13 shows a wavelet coefficients vector of details. All elements of this vector should be zero. Due to the interference of the other sub-bands, the vector is modified. But the values of the elements are still very small, and this can help us to determine the watermark.

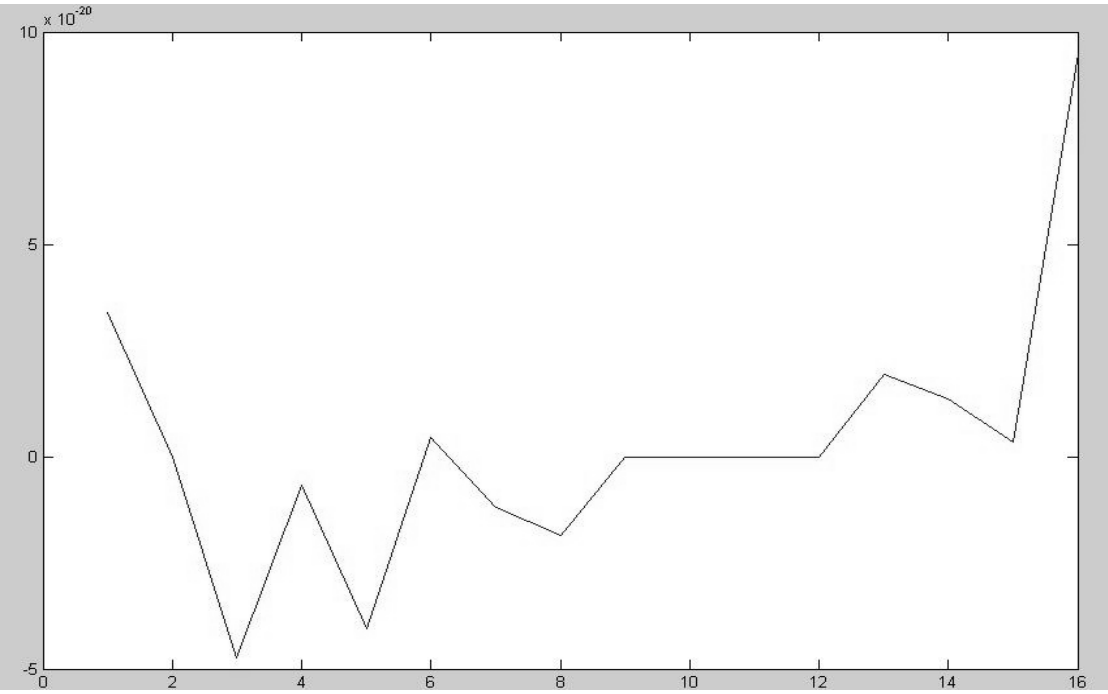


Figure 4.13: The Extracted Watermarked cDD

Table 4.2: Error Rate

	Original	**Adding 2% Gaussian noise**	**Adding 5% Gaussian noise**
"clar.wav"	0.0084	0.0374	0.0967
"att_01.wav"	0.0191	0.0627	0.1139

From Table 4.2, we can find the proposed algorithm is sensitive to the noise. Since the stochastic part of an audio signal is noise like, and the watermark is embedded in the stochastic part. Thus introducing the noise can degrade the performance of the algorithm dramatically.

4.4 A Speech Watermarking Algorithm Incorporating with GSM 610 Speech Coder

Linear prediction is a method for the discrete time signals, and it predicts the future values based on a linear functions of former samples sample of the signal. A filter transfer

function is exploited to simulate the vocal tract transfer function. By ingenious design, the linear predictive coding can yield high quality of speech.

4.4.1 GSM 610 Speech Coder

Speech coders can be classified into three types: Waveform coder, source coder and hybrid coder. The waveform coders attempt to maintain the original waveform and the coding is based on quantisation and redundancy within the waveform. In contrast, source coders make no attempt to reproduce the original waveform, but instead derive a set of parameters at the encoder, which are transmitted and used to control a speech production model at the receiver. Hybrid coders combine features from both waveform coders and source coder to provide good quality, efficient speech coding. At rates between about 16 kbit/s and 4 kbit/s, good quality coding is achieved using 'analysis by synthesis' techniques, as shown in Figure 4.14. The objective is to derive an excitation signal, like that produced by the glottis, such that the difference between the input and synthesised speech signals is minimized according to some suitable perceptual criterion. GSM 610 (RPE-LTP) coder [4.14] is a kind of widely used hybrid speech coders .

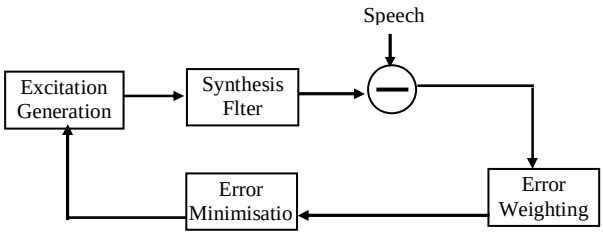


Figure 4.14: Analysis-by-synthesis Coding Scheme

Regular Pulse Excitation, Long-Term Prediction (RPE-LTP) linear predictive coders [4.14] are widely used in current wireless telecommunication. We can find them in cell phones and PDAs. It first removes local correlation between samples and produces either a pitch residual, or a noise-like signal for unvoiced speech, and then quantifies the harmonics of the speech. Later, the RPE stage reduces the residual samples down to four sets of 13-bit sub-sequences. The optimum sub-sequence is determined as having the least error. The resulting signal is fed back in order to help the processing of the next frame. The resulting signal is shown in Table 4.3.

Table 4.3: Output of RPE-LTP Vocoder

Parameter	**Number of values**	**Bits per frame**
LARs	8 per frame	36 bits
LTP lag	1 per subframe (7 bits)	28 bits
LTP gain	1 per subframe (2 bits)	8 bits
RPE grid position	1 per subframe (2 bits)	8 bits
Block amplitude	1 per subframe (6 bits)	24 bits
RPE Pulses	13 per subframe (3 bits each)	156 bits
Total		**260 bits**

4.4.2 Speech Feature Extraction for Efficient Authentication

Feature extraction method is widely deployed for speech integrity authentication algorithms. Some audio watermarking algorithms use different statistical properties, such as pitch values, salient points and so on, of the host audio and modify them in order to embed watermark. Paper such as [4.12] introduced content-adaptive segmentation of the host audio according to its characteristics in time domain. The algorithm presented in [4.13] extends the mute period of the host audio to embed the watermarks. The technique of self-embedding, illustrated in Figure 4.15, is a kind of feature extraction skill, which tries to embed extracted feature values back into the audiovisual data. Wu's work [5.5] shows that self-embedding has two drawbacks: (a) a very high data capacity for watermarking; (b) most self-embedding methods operate at high computational costs. Thus the known self-embedding algorithms are not applicable for low data rate speech encoders such as GSM 610 voice encoder. Therefore, we developed a new feature-embedding algorithm for such encoders.

Two design requirements for speech feature selections include:

1) In order to resist to the lossy speech coding, the selected features should not be altered greatly by speech coding;

2) The computation cost of feature extraction should be small.

Since no computationally demanding transform of the host signal needs to be done, an algorithm operates in time domain has a very small algorithmic delay. This permits the use on this kind of algorithms in real-time applications.

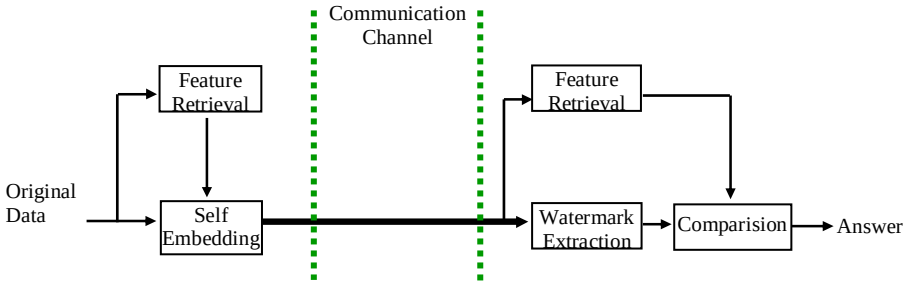


Figure 4.15: Self-embedding Scheme

4.4.3 Watermark Embedding and Extraction

In real-time speech processing the computation and delays of the watermarking algorithm should be small enough. In speech coding, for example, due to the transmission of speech, other resources of the system also become more limited. In addition to the computational complexity and delays, the bandwidth sets its limitations to the amount of data transmitted, i.e. to the number of parameters used and to the accuracy of quantization applied. This requirement makes the voice encoders eliminate the redundancy of the speech and minimize the volume of the quantization coefficients.

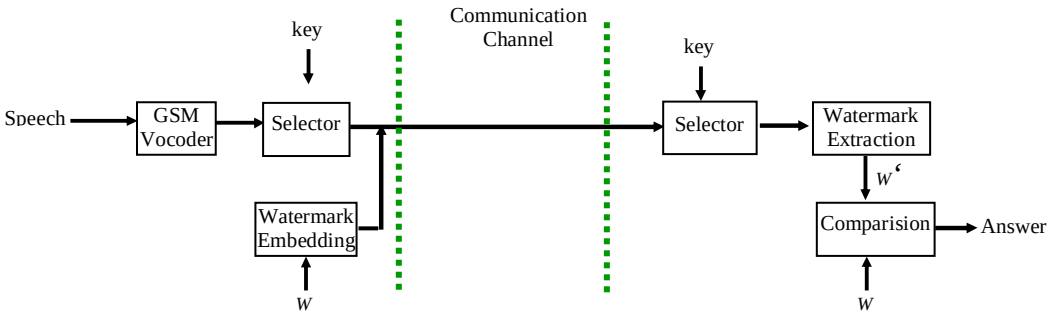
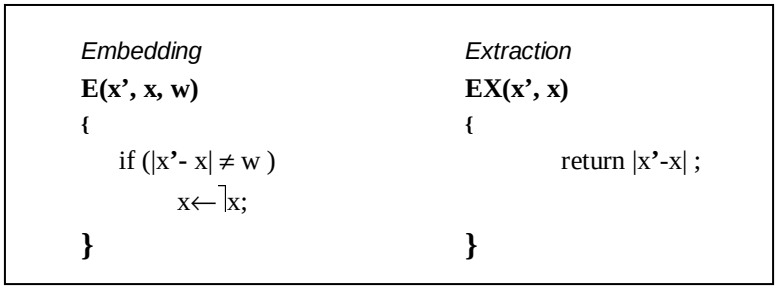


Figure 4.16: Proposed Speech Watermarking Scheme

In the proposed scheme, RPE pulses are modified slightly to engrave the digital watermarks. In this work, two digital watermarks, namely *speech_watermark* and *comm_watermark*, are used for each transaction. For the purpose of source origin authentication and integrity, a speech message will be inserted into a unique fragile watermark, *speech_watermark*. In contrast, a *comm_watermark* will be inserted into the

speech message to ensure the authenticity and integrity of real-time speech transmission. Both digital watermarks and private keys are binary sequences.

A GSM 610 coder generates 52 RPE pulses of 3 bits each for a speech frame of 20 ms duration. An RPE pulse pair in one frame is chosen to embed one bit of the digital watermark. The pulse couple is selected according to *key*, as depicted in Figure 4.16. Consider x' and x are the least significant bits of the selected candidate RPE pulses. Then the watermark embedding and extraction algorithm works as follows:



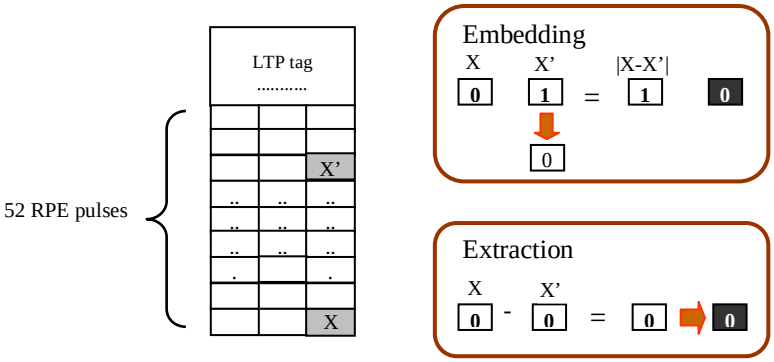


Figure 4.17: Audio Watermarking Algorithm

(1) *speech_watermark*

key is an integer pair {(p1, p2) | 0≤ p1, p2<50}. To embed one bit of the secret watermark, *speech_watermark*, select two RPE pulses from the first 50 RPE pulses according to *key*. The least significant bit of two selected RPE pulses are the candidate bit pair (x, x'). Apply the algorithm mentioned above to engrave a bit of *speech_watermark*.

(2) *comm_watermark*

comm_watermark is public, the last two RPE pulses of one frame are selected to hide one bit of *comm_watermark*. The least significant bit of each selected RPE pulses are the candidate bit pair (x, x'). Apply the algorithm mentioned above. Figure 4.4 shows an example of the proposed audio watermarking algorithm.

The proposed algorithm uses feature extraction technique to enhance the authentication capacity. Each one identifies oneself by his unique speech. Log Area Ratios (LARs) are the reflection coefficients modeling the shape of vocal tract, thus that it is very common-used in speaker recognition techniques. So we choose LARs as the principal speech features.

GSM 610 vocoder generates 36 bits LAR parameters for every 20ms, and its feature value can be coarsely calculated as:

$$F(LARs) = \mathrm{MOD}\left(\sum_{i=1}^{36} LAR_i \right) \qquad (4.6)$$

Then apply $\Theta(w, F(\cdot))$ to generate a new watermark bit *w'*. Hence the feature of vocal tract is integrated into the proposed speech watermarking algorithm.

Table 4.4: $\Theta(w, F(\cdot))$

		F(LARs)	
		0	**1**
W	**0**	**1**	**0**
	1	**0**	**1**

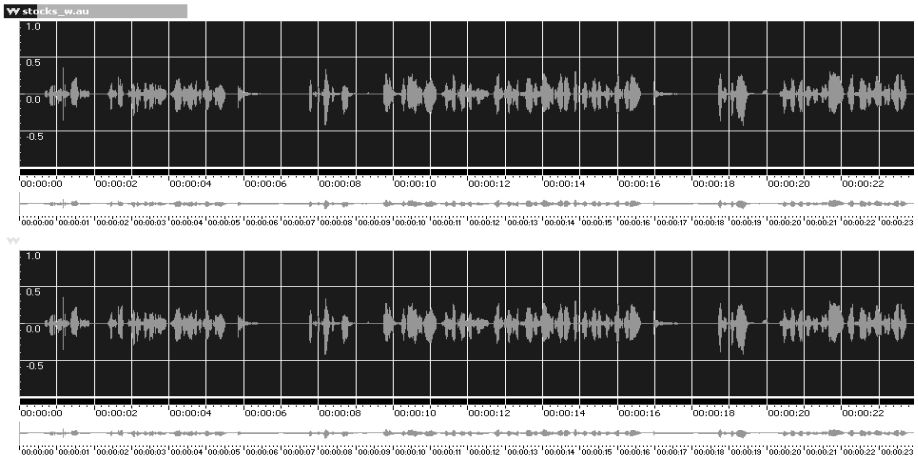
4.4.4 Experimental Results

The overhead brought on the GSM vocoder by the watermarking module is shown in the following table, whereas a Pentium 3 – 800MHz CPU with 356MB RAM was used.

Table 4.5: Watermarking Overhead on GSM 610 Coder

Encoding/decoding Time of a Frame	**Computing Time for Watermarking in a Frame**	**Watermarking Time/Encoding Time**
287.63 µs	2.79 µs	1 %

The distortion introduced by the watermarking is rather low, see Figure 4.18. It is very clear that this speech watermarking algorithm has a quite limited usage. But the similar approach can be used in watermarking algorithms for the other vocoders deploying linear predictive coding, such as CELP vocoder.



(a)

(b)

Figure 4.18: Watermarked (a) and Original Speech (b)

Chapter 5 Watermarking Mechanism For Real-Time Speech Transmission

Internet telephony unveils some special properties stemming from both multimedia data and real-time communication. First, end users cannot perceive limited distortions in multimedia data, so, some bit errors and packets losses occurring during communication do not defect the overall visual/audio quality. Secondly, due to controlling protocols and implementations of real-time multimedia communication, packet losses may happen any time. Thirdly, caused by the large amount of multimedia data, the communication security trade-offs should be as low as possible. In this chapter, we present a security scheme using speech watermarking technique.

5.1 Real-time Speech Communication Systems

VoIP or Internet telephony is a technology allows people to perform telephone service using the existing IP-based data network, thus it can avoid the tolls charged by telephone service provider. VoIP booms in the past five years. Skype and Vbuzzer are the best runners in the competition. However, lot of security incidents occurred in the last years and their number is still increasing. Therefore, security is an essential topic for VoIP telephony systems [5.2].

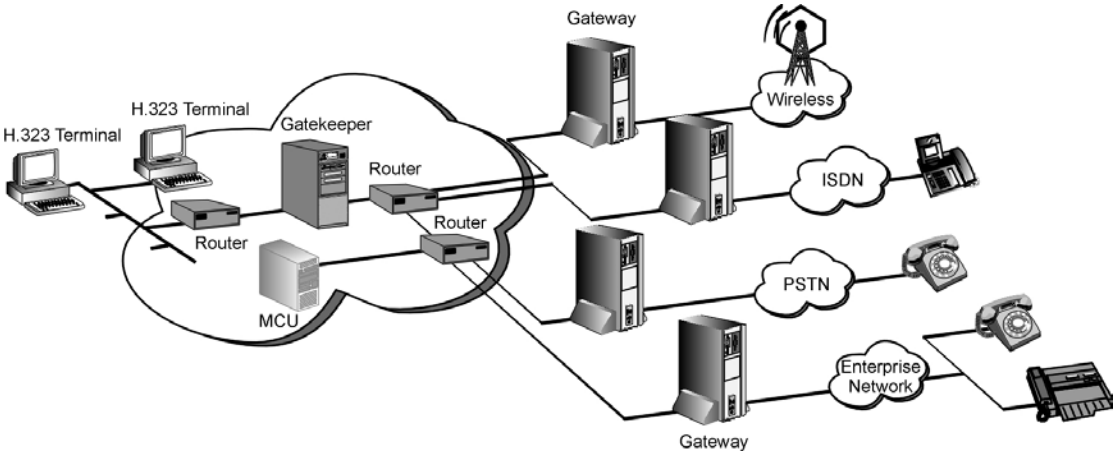


Figure 5.1: A Typical VoIP Implementation

Source: http://www.ebstrategy.co.uk/whitepapers/voip/voipsolutions.pdf

5.1.1 Architecture

A typical VoIP network is illustrated in Figure 5.1. A VoIP network has the major components similar in functionality to a traditional circuit-switched network. There are three major components of a VoIP network, and they are voice gateways, signaling controllers and IP network. The most important techniques to VoIP includes high quality low-rate voice coders, routing techniques and transmission protocols.

5.1.2 Major Transmission Protocol and RTP

There are four main stream transmission protocols being used nowadays. The four transmission protocols are originated from the different sources or different communities. H.323, SIP (short for Session Initiation Protocol) [5.14], MGCP (short for Media Gateway Control Protocol) [5.15] and Megaco (short for Media Gateway Control Protocol) [5.12] are the VoIP protocols suites standardized by the different organization or vendors. Among them, H.323 and SIP are the most widely used. H.323, developed by ITU in 1996, enables voice communication, video communication and files transmission. SIP, defined by IETF, is a signaling protocol and works on application level. SIP can establish a VoIP connection and supports IP telephony, conferencing and instant messaging.

In a packet-switched network, TCP is used for signaling, parameter negotiations, path setup, and control. UDP is used for transmission of payload (traffic) from sources generating real-time packet traffic.

A note worth to be mentioned is that SIP can perform the data transmission on HTTP protocol. This is a big attempt to increase the flexibility of VoIP systems to transmit data over heterogeneous networks. SIP can deploy a lot of low cost devices, such as cell phones, network computers, and set-on-top boxes. In addition, it is easy to extend the SIP protocol family to perform the new service.

The Real-time Transport Protocol is defined as a transmission protocol, which provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video. Both H.323 and SIP utilize RTP protocols as the carrier protocols.

5.1.3 Security Issues of VoIP

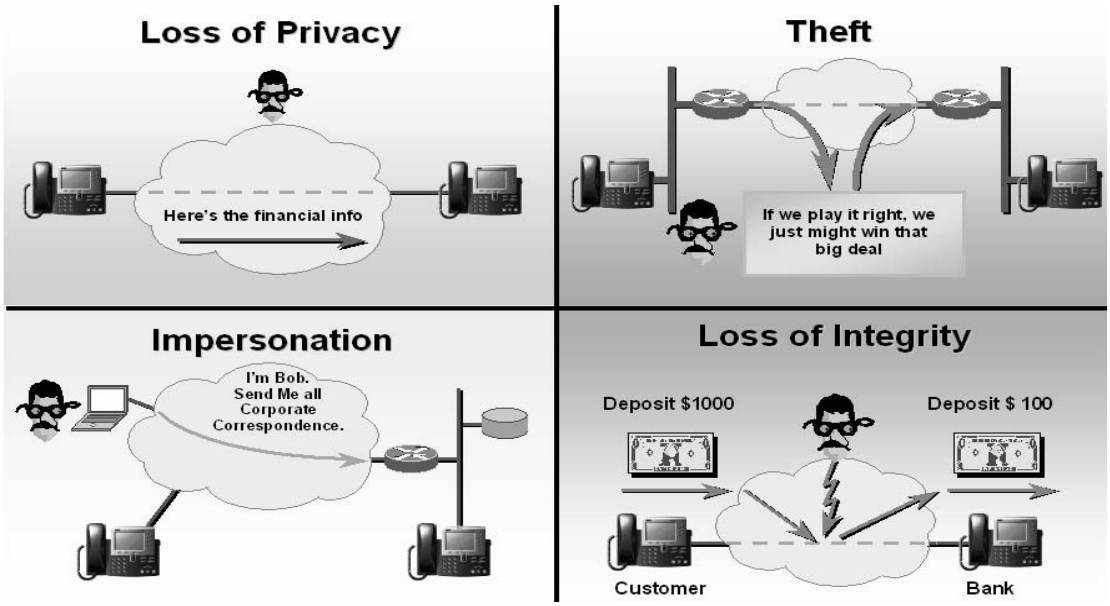


Figure 5.4: Security Threat to VoIP Systems

Source: "Enterprise IP Telephony Securing Voice Today", Cisco, 10/6/2003

There are five different categories of security services present for IP telephony: Identification and authentication, Authorization, Confidentiality, Integrity, Non-denial/Non-repudiation. These security aspects can be addressed by means of different cryptographic techniques such as secret key cryptography, public key cryptography, and hash functions, respectively. In this work, we focus on integrity and source origin authentication of multimedia data. By multimedia data integrity and authenticity mechanisms, we can provide some very useful related information, such as the speaker's identity, the point in time when dialog happened, and the integrity/authenticity of the transaction.

There are two types of authentication: Proof that the packets have not been tampered with, known as *integrity protection*, and proof that the packets came from the correct source, known as *source origin authentication.*

Integrity protection is achieved through the use of message authentication codes. These codes take a packet to be protected, a key known only to the sender and receivers and use these to generate a unique signature. Provided that the key is not known to an attacker, it is impossible to change the contents of the packet without causing a mismatch between the packet contents and the message authentication code.

Source origin authentication is much harder for RTP applications. It might be simply addressed by means of message authentication codes, but it is more difficult because a shared

secret between sender and receiver is not sufficient. On the other hand, the digital signature is larger and needs a high computing resource for public key cryptography. In addition, the source origin authentication is only applicable over a single RTP data packet. There is no existing solution to ensure the content security of the whole session.

Like confidentiality, authentication can be applied at either the application layer or the IP layer, with much the same set of advantages and disadvantages. Both alternatives have been developed for use with RTP.

Standard RTP [2.11] provides no support for integrity protection or source origin authentication. Implementations that require authentication should either implement secure RTP or use a lower-layer authentication service such as that provided by the IP security extensions.

i) Authentication Using the Secure RTP Profile

SRTP [2.1] supports both message integrity protection and source origin authentication. For integrity protection, a message authentication tag is appended to the end of the packet, as it was discussed in Section 2.1.2.2. The message authentication tag is calculated over the entire RTP packet and is computed after the packet has been encrypted.

ii) Authentication Using IPSec

Authentication in IPSec is provided by the Encapsulating Security Payload profile, or Authentication Header (AH) [2.2].

iii) Confidentiality Using IP Security

IPsec [2.2] provides confidentiality by encrypting the payload of RTP data packets but it is not RTP-specific.

iv) Fast Encryption Methods for Audiovisual Data Confidentiality

Selective algorithm [2.3], ZIG-ZAG permutation [2.4] and Video Encryption Algorithm [2.5] incorporate cryptographic techniques with multimedia compression algorithm.

v) Digital Watermarking

Usually, the sender applies MAC or digital watermarking on the multimedia data to achieve the integrity and authenticity. As well, the receiver applies MAC or watermarking algorithms for the future integrity and authenticity check or multimedia indexing [5.11].

The existing security approaches work in different levels/components in secure multimedia communication systems. Fast encryption methods for multimedia data (i.e., Selective algorithm [2.3], ZIG-ZAG permutation [2.4] and Video Encryption Algorithm [2.5]) incorporate cryptographic techniques with multimedia compression algorithm. SRTP, which provide encryption and integrity functionalities, works in application level in TCP/IP stack. TLS/SSL and IPSec can set up secure channels between the sender and the receiver and they work in the transport layer and network layer respectively.

5.2 Watermarking-Based Security System for Real-time Speech Communications

A large variety of audio watermarking algorithms has been proposed in the past and a few of them can be adapted for speech watermarking applications such as [5.5, 5.6, 5.7]. Many of these algorithms operate in time domain and exploit temporal or spectral masking models of the human auditory system. The most challenging requirement in real-time application, however, is to detect the fragile watermark in, say, every 0.5 seconds of speech with a reasonable error probability, especially when modern speech coders compress 0.5 sec of speech to less than 400 bytes of data.

The digital watermarks embedded in the original data may contain some useful information, e.g., author names, date of generation, or copyright holders. With the use of the blind digital watermarking algorithm, the embedded digital watermark can be extracted accurately without the need of the original multimedia data. Potential attackers, on the other hand, cannot retrieve the embedded watermarks without the knowledge of the private key or of the original data, respectively. Fragile audio watermarking algorithms can thus detect severe tampers/attacks occurred on the multimedia data. Therefore, it is a pretty useful method to ensure both authenticity and integrity of multimedia data.

Fragile digital watermarking provides an alternative approach to increase the safety of multimedia data during transmission in openly accessible channels. That is, digital watermarks may be generated referring to information on, say, originators, receivers, unique serial number, and time stamps. These watermarks are then embedded into the multimedia data to assure its integrity and source origin authentication without degrading the overall quality of the transmitted multimedia data.

A point should be mentioned is that integrating speech watermarking technique with VoIP can solve the authentication requirement and integrity requirement of an Internet telephony system. The confidentiality or secrecy should be achieved by encryption. An alternative for the encryption scheme is to exploit the audio or video perturbation. Multimedia data perturbation is a slight work load secrecy method with promising future.

5.2.1 Outline

SRTP protects the integrity of each data packet instead of the whole conversation data which consisting of hundreds or thousands of audio data packets. In Internet telephony, an audio data packet contains the compressed sample for short 20 ms, thus a packet loss does not degrade the quality of service. Figure 5.5 is an integrity model applicable for the whole conversation data. Each speech data packet, p_k, has its own local integrity value, I_k, and the overall integrity is denoted as I. The overall integrity function is defined as

$$I = \frac{\sum_{i=1}^{N} I_i}{N} \qquad (5.1)$$

here N is the total number of the packets, and $\{ I_k \mid 0 \le I_k \le 1, 0 \le k \le N\}$

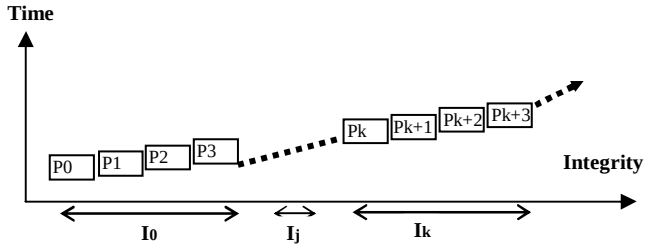


Figure 5.5: A New Integrity Model

Usually, the Internet telephone sends an RTP packet at an interval (denoted as t) of 20ms or 40ms, and 0.5s of speech (denoted as s) represents one accurate semantic meaning, so that the distortion of a RTP packet in a short period does not change the semantic meaning of the whole session. In other words, if a burst of distortion lasts for less than 0.25 s or for $s/2t$ RTP packets being transmitted, then the quality of the session does degrade, but the integrity or authenticity of the multimedia communication is not destroyed. So, the proposed integrity and authenticity measurement can be taken on a range consisting of m ($m << s/2t$) RTP packets to tolerate the possible network distortion.

In our work, a local integrity value, I_k, is calculated at intervals of 12 RTP packets. With the help of sequence number of the received packets, the synchronization of assessment can be easily achieved. Due to the packet loss, in some cases only p ($p < 12$) RTP packets are received in an interval. If the number of lost RTP packets is small, say, less than 2, we can ignore the small distortion and we set the local integrity value to 1.

Digital watermarking provides an alternative approach to ensure the safety of multimedia data during transmission in openly accessible channels. The digital watermarks may be embedded into the multimedia data to assure its integrity and authenticity without degrading the overall quality of the transmitted data.

We apply the scheme described in Section 2.2.1. In the real work, both the sender and receiver share one reference watermark (*comm_watermark*) from an independent trustworthy party before the multimedia data transmission really starts. To provide the non-repudiate and authenticity, a secret digital watermark, *speech_watermark*, is being introduced. *speech_watermark* has previously been encrypted with a public key algorithm, e.g., RSA.

The sender embeds both a public, *comm_watermark*, and a secret watermark, *speech_watermark*, into the outgoing multimedia data stream. The receiver then extracts the public digital watermark from its incoming stream. This scheme provides an integrated solution to secure multimedia communication and multimedia data integrity.

The network transportation path can be viewed as a noisy channel. The reference digital watermark is modulated with multimedia data (carrier) and transmitted onto the noisy channel. The watermark undergoes the same changes suffered by the multimedia data, so that the watermark degradation can be used to estimate the overall alterations of the multimedia data caused by noise or by attacks. At the receiver side the embedded digital watermark is extracted and compared to the original reference watermark in order to measure the integrity and authenticity of the received multimedia data.

5.2.2 Suppression of the Packet Loss

Packet losses always happen during the network transmission. This makes it very difficult to devise objective quality measurements for different repair schemes. It is not sufficient to measure the difference between the original waveform from the source and the waveform recovered at the receiver, because the perceived quality has no direct relation to the differences in the waveforms.

The packet loss can be determined by checking the sequence number of a RTP data transfer packet. The receiver has a ring to record the sequence numbers. The receiver extracts

the sequence number of each incoming RTP data packet and then writes it to the ring. If the newest entity of the ring is not adjacent to its nearest entity, which shows RTP data packet has happened.

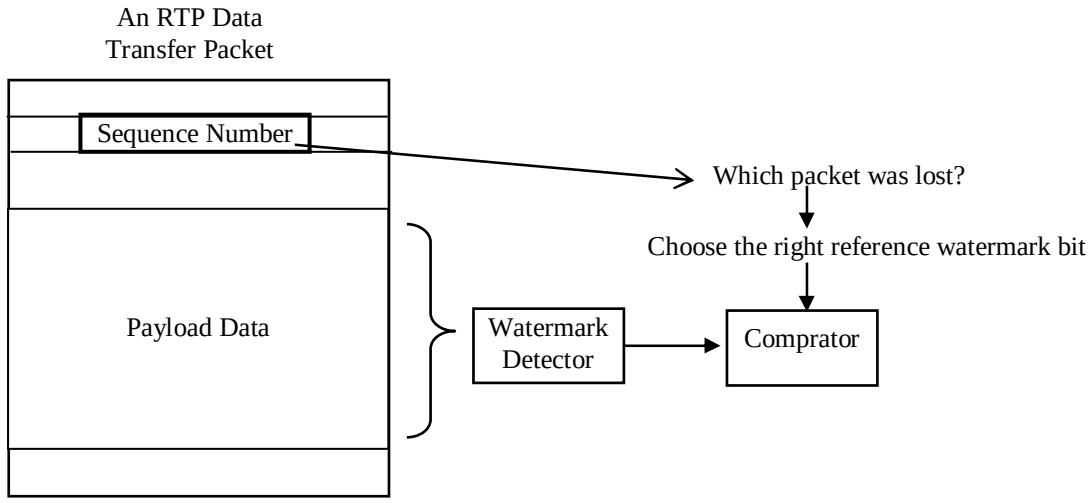


Figure 5.6: Packet Loss Suppression

Since packet loss always happens during the transmission, synchronization between sent and received speech signals are very important to this approach. If some RTP data packets were lost, the watermark detector should skip the number of lost data packets to synchronize the watermarking authentication procedure.

5.2.3 Operation

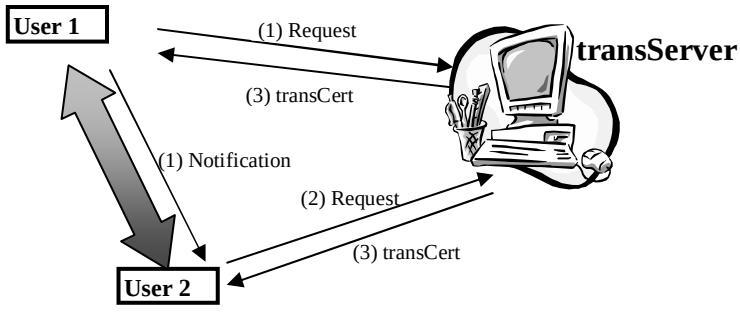


Figure 5.7: Secure Multimedia Communication

At the beginning of each real-time multimedia communication, user2 sends a *dialog_start* requests to *transServer* and notifies the callee, user1. Then, user1 also sends a *dialog_start* requests to *transServer*. The *dialog_start* requests contain the digital certificates of the communicating parties. *transServer* authenticates the participants by verifying their digital certificates extracted from the received *dialog_start* requests.

If acceptable, *transServer* generates a pair of *transCert*s, sends the *transCert*s to the participants, and stores them in its database, as depicted in Figure 5.7.

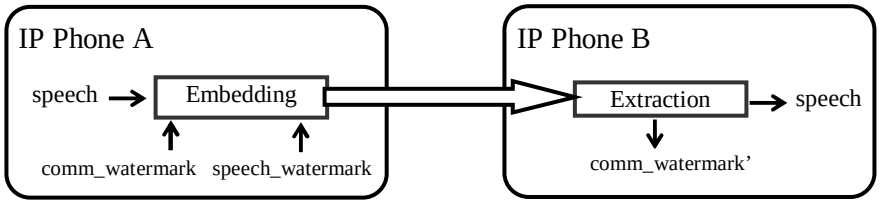


Figure 5.8: Real-Time Authentication for VoIP

Once the participants receive the *transCert*s from *transServer*, the conversation may begin. At first, they authenticate the received transaction certificate. If the authentication is successful, then the participants parse the received *transCert*s to ((*speech_watermark, key*) *comm_watermark*) and insert then the digital watermarks into their outgoing audio data streams. At the same time, both participants extract the *comm_watermark* from their incoming streams and authenticate them, as depicted in Figure 5.8. At the end of the transmission, i.e., conversation, each of the participants sends a *dialog_end* request to *transServer*. *transServer* authenticates the received *dialog_end* request and puts them on file.

The simplest way to tolerate packet loss during the real-time transmission is present in 5.2.2. But in order to enhance the tolerance to packet loss and bit errors that may occur in the saved multimedia data, a more flexible solution is given. The sender deploys the derived Walsh codes to modulate the digital watermarks. Some synchronization marks, denoted as SYNC, are inserted into the modulated Walsh codes sequence. That is 1111 stands for SYNC, 1001 stands for '0', and 1010 stands for '1'. The resulting sequence is represented as $\{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7,$ SYNC, ….., SYNC, $b_{j-1}, b_j, b_{j+1}, \ldots b_n\}$. The bits in between an adjacent SYNC pair consist of a phase. The sender embeds this bit sequence into the outgoing multimedia stream.

On the other side, the receiver retrieves a bit sequence, {b_0', b_1', b_2', b_3', SYNC, ….., SYNC, b_{j-1}', b_{j+1}',….b_n'}, from the incoming stream. Since both the sender and the receiver share *comm_watermark*, the reference bit sequence, {b_0, b_1, b_2, b_3, SYNC, ….., SYNC, b_{j-1}, b_j, b_{j+1},….b_n}, is accessible by the receiver.

During the communication procedure the receiver performs the following steps to evaluate the integrity and authenticity of the data being transmitted in the channel:

Initialization: prepare a buffer to hold the retrieved bits
Step:
1) scan buffer; if SYNC found, then goto step 1
2) extract one embedded digital watermark bit from one multimedia frame and put it into buffer; repeat step 1
3) evaluate the partial integrity and authenticity; clear the buffer
4) if not end of communication, then goto step1.

Due to the possible packet loss and bit errors, the integrity and authenticity assessment looks a little complicated.

(1) If the size of the extracted bit sequence (N) is equal to the size of the reference bit sequence (no packet loss has happened), then compare the extracted bit sequence with the reference bit sequence as follow:

$$I = \frac{\sum_{i=1}^{N}(b_i \text{ XNOR } b_i')}{N} \qquad (5.2)$$

(2) Maybe the size of the extracted bit sequence, M, is not equal to the size of the reference bit sequence, N, either due to packet loss or due to a loss of SYNC. If the size of the extracted bit sequence is smaller, then the integrity and authenticity estimation work as follows:

Denote the reference bit sequence as B_r and the extracted bit sequence as B_x
Step:
1) find the most left and the most right common strings (sizes are denoted as l_l and l_r) from B_r and B_x
2) remove the most left and the most right common strings from the two bit sequences

3) find the longest common string from the two bit sequences; the length of the common string is l_c

The integrity value of this phase is given by

$$I = \frac{l_l + l_r + l_c}{N} \tag{5.3}$$

(3) If the size of the extracted bit sequence is larger (the synchronization code has been destroyed), then combine the adjacent two reference phases and perform the above procedure to estimate the multimedia integrity and authenticity.

Many existing real-time multimedia communication protocols deploy symmetric ciphers to set up a secure channel. In these cases the proposed real-time integrity and authenticity assessment can be exploited as a method to measure the quality of the transmission.

5.2.4 Source Origin Authentication

The secret digital watermarks, *speech_watermark*s, embedded in the multimedia messages have been signed by *transServer* using public-key encryption. This means that each of the watermarked conversation data is distinguishable from others. Therefore, one can verify the source origin of the saved multimedia data: 1) A (*speech_watermark, key*) pair can be parsed from the proper *transCert*; 2) extract a digital watermark, *w*', from the recorded conversation data and compare it with the original one to authenticate the inspected speech. The authentication procedure can be illustrated by pseudo codes as follow:

```
(speech_watermark, key) ← Parse( transCert )
Answer ← Authen( speech_watermark, PK )
if ( Answer ≡ TRUE ) {
        w' ← Extract( Recorded Dialog, key )
        Answer ← Compare(w', w )
}
```

5.3 An Implementation on Internet Telephony

5.3.1 Components of Test Environment

To prove the evidence of the proposed scheme, a prototype has been developed and implemented by means of a desktop computer and a notebook. The prototype architecture consists of a pair of IP phones and a *transServer*.

IP Phones

IP phones are adapted from Speakfreely [5.8], a well-known open source IP telephony implementation. Both the implemented *transServer* and IP phones employ the OpenSSL toolkit to handle digital certificates and to generate *transCert*s.

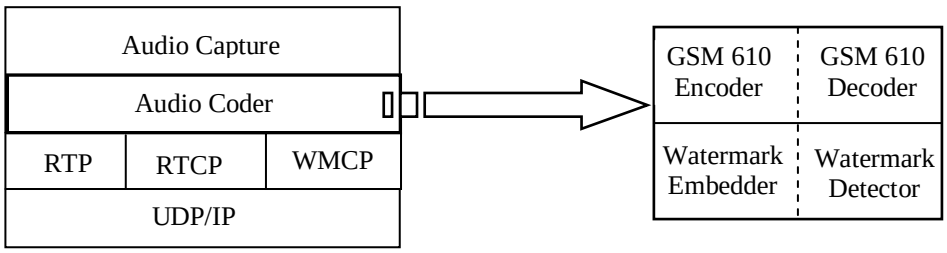


Figure 5.9: Software Components of an Internet Telephone

Each IP phone has been assigned a X.509 digital certificate to uniquely identify itself. All of the IP phones' digital certificates are registered by *transServer*.

Watermarking Control Protocol (WMCP) is an author-designed software component, which is in charge of digital watermark embedding/detection, set-up process control. Also the last procedure of speech coding of GSM 610 vocoder has been modified.

In the beginning of the conversation, the WMCP uses the phone's certificate to generate a *Dialog_start_request*, and sends the request to the *transServer* via the OpenSSL enabled secure channel. Once receives the answer from the *transServer*, the WMCP validates the received *transCert* first, and parse the certificate to generate a private watermark, *speech_watermark*, and a public watermark, *comm._watermark*. Then WMCP activates the audio capture unit of the IP phone, and sends the watermark pair to the GSM encoder/decoder. The watermark embedder inserts the watermark pair into the frames, and the detector extracts the embedded public watermark bits from the incoming audio frame. The operations of WMCP can be illustrated as follow:

Phone

Setup a secure channel with *transServer*

Rq ← (Phone's Certificate, TID, RQ_FLAG)$_k$ // Generate Dialog Request

Dialog_Start_Rq → *transServer* // Send request to *transServer*

// Waiting

Authen((DigCert, TS)$_k$)$_k$

Open two secure channels with the other phone

//(a signaling channel and a data channel)

w ← Parse(transCert)

{

on data channel

forever

{

{

Embed(voice frame, w') → watermarked frame

watermarked frame → another phone

}

{

receive(watermarked frame)

extraction(watermarked frame) → w

}

}

on signal channel

waiting

if (controlling signal ≡ TERMINATE)

{

break the data channel

Dialog_end_rq → *transServer*

}

}

transServer

transServer is a server facilitated with OpenSSL toolkits. It authenticates the IP phones' identifications during the setup of a conversation, generates a pair of *transCert*s, and

issues them to the appropriate IP phones. At the end of the conversation, it receives the request from the IP phones, and then finishes the conversation. *transServer* uses *service_no* to identify the multiple conversations happening in parallel.

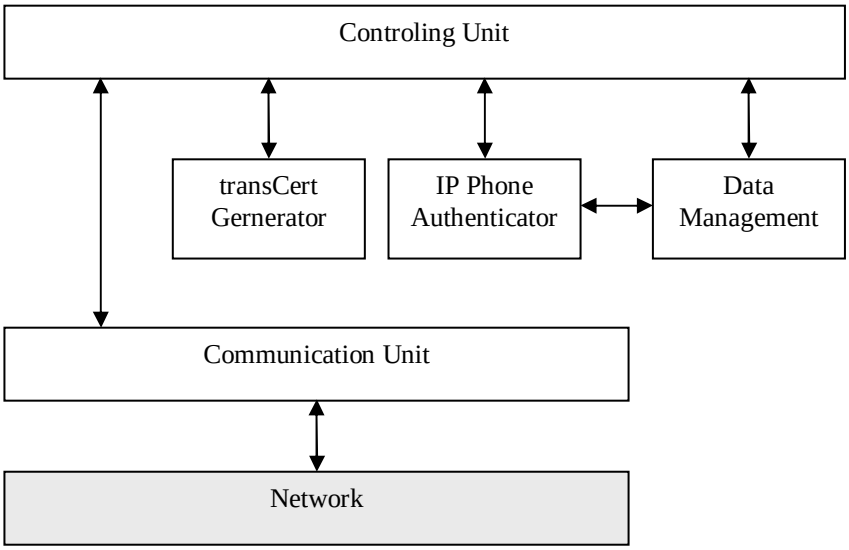


Figure 5.10: Software Architecture of *transServer*

Fig. 5.10 illustrates the layered architecture of *transServer*. The controlling unit is monitoring the running processes of the whole server. It communicates with the communication unit, *transCert* generator and IP phone authenticator. The communication unit is listening to the connection port all time. And it transmits the request to the controlling unit. Once the controlling unit receive a request, it analyses the request and decides which software modules should be called. Specifically, if the controlling unit receives a *Dialog_start_request*, it retrieves the certificate of the caller, and sends the certificate to the IP phone authenticator. If the return from the phone authenticator is true, the controlling unit generates a *service_no*, and sends the *service_no*, the caller's ID and callee's ID to the *transCert* generator. The *transCert* generator generates a *transCert*, and signs the private watermark, *speech_watermark*, with the private key of the *transServer*. Once the controlling unit received the returned *transCert* from the *transCert* generator, it put the *transCert* on the transaction list and sends the *transCert* to the participators of the conversation.

If the controlling unit receives a *Dialog_end_request*, it inserts a timestamp to the transaction list to indicate the time of finish and then moves the record, which containing the information of this transaction, from the transaction list, into data management unit. At last it

sends the answer to the participators to finish the conversation. The OpenSSL utilities are utilized for several times during one transaction.

The operations of the *transServer* can be illustrated as follows:

transServer

```
                                        // Waiting
        Analyse the received request
        If (Dialog_Start_rq ≡ TRUE )
        {
                Authen( Phone's Certificate) →isValidPhone          // Authenticate Phone
                if ( isValidPhone ≡ TRUE )
                {
                        watermark, key ← RANDOM()                   // Generate a transCert
                        tag ← ( SessionID, RAND )
                        transCert ← Sign( tag, PK )
                }
                (time mark, caller's ID, callee's ID) →Database
                transCert→Phones                                    // Send a transCert to the phone
        }
        If (Dialog_End_rq ≡ TRUE )                                  // at the end of a conversation
        {
            Authen( Phone's Certificate) →isValidPhone              // Authenticate Phone
                if ( isValidPhone ≡ TRUE )
                {
                        (duration, isSuceed) ← Dialog_End_rq()              // Generate a transCert
                        (duration, time mark, isSuceed, service num) →Database
                }
        }
                                        // Waiting
```

transCert

A pair of *transCert*s is issued by *transServer* to identify this conversation. A *transCert* contains a private watermark, its key, a public watermark and a service registration number.

```
struct trans_cert
{
        unsigned char   speech_watermark[32];   // private watermark
        unsigned char   key[256];               // key for private watermark
        unsigned char   comm_watermark[16];     // public watermark
        long            service_no;             // conversation registration number at transServer
};
```

5.3.2 Experimental Results In Local Area Network

Some test runs for the proposed speech watermarking algorithm were performed. The first experiment was performed on a 100M Ethernet LAN being used as the communication network. In addition, we tested the network scenario comprising both an 100Mb/s Ethernet LAN and a 11Mb/s WLAN. The upper row of Table 5.1 shows the result of a test, which was run by a VoIP end-point couple on 100Mb/s Ethernet. The lower row shows the test result taken from the wireless LAN scenario as described in Section 1. The average packet loss (including packet delay) is quite low for VoIP devices working on wired LAN and the performance decreases in a wireless environment. But the proposed integrity measurement is robust enough to suppress the degradation of the network.

Table 5.1: Integrity Values in WLAN

	No. Total Packet	**No. Packet Loss, Delay & Errors**	**No. Water-mark Errors**	**Integrity Value**
Test on 100Mb/s Ethernet LAN	16390	6	6	99.96%
Test on 11Mb/s WLAN	16908	285	112	99.83%

To prove the evidence of the proposed scheme, a prototype has been developed and implemented by means of desktop computers. The prototype architecture consists of a pair of IP phones and a *transServer*. IP phones are adapted from Speakfreely, a well-known open

source IP telephony implementation. Both the implemented *transServer* and IP phones employ the OpenSSL toolkit [5.9] to handle digital certificates and to generate *transCert*s. A desktop and a notebook are used in our experiment as the IP phone devices. The desktop has a Pentium 3 800MHz CPU and 356 MB RAM, whereas the notebook features a Pentium Celerion 2GHz CPU and 256 MB RAM. The experiment was performed on a 100M Ethernet LAN being used as the communication network. IP phones can perform the watermarking operation very quickly as shown in Table 5.2.

Table 5.2: Integrity Values

	No. Total Packet	**No. Packet Loss& Delay**	**No. Water-mark Errors**	**Integrity Value**
Test 1	16390	6	6	99.96%
Test 2 (on busy LAN)	15952	48	32	99.89%

Table 5.3: Time Cost of *transCert* Operation

Time Cost of *transServer*'s Response Procedure	**Time Cost of Internet Telephone's Acquisition Procedure**
2876.33 ms	5935.85 ms

At the beginning of a conversation, each participating Internet telephone asks for a *transCert* from *transServer*; the corresponding *transServer* authenticates the participants, generates a pair of *transCert*s and distributes them to the participating Internet telephones. The Internet telephone's *transCert* acquisition procedure begins with sending a request to *transServer* and ends with receiving a *transCert*; *transServer*'s response procedure begins with receiving a request, and ends with the *transCert* pair distribution. Due to the requirement of real-time communication, both the Internet telephones' acquisition procedure of *transCert* and *transServer*'s response procedure should work very efficiently. As shown in Table 5.3, the average processing time consumed by the *transCert* operation is well acceptable for both Internet telephones and *transServer*.

Figure 5.11: Security System Using Watermarking

5.3.3 Experiments In Wireless Local Area Network

At present IEEE 802.11 wireless LANs [5.1] are spreading very fast. Applying Internet Telephony techniques over WLANs may be a low-cost voice communication means within enterprise or campus networks. VoIP over WLAN is becoming an important short-range communication method. Before that can happen, however, two technical problems need to be solved. The first is that the system capacity for voice can be quite low in WLAN, and Table 5.4 shows some packets are lost in presence of heavy load (> 4Mbps of background traffic). The second is that TCP-based applications can bring down VoIP performance.

Table 5.4: Packet loss in WLAN

Packet size	120	160	500	1000
Packet loss rate	22.9	14.6	3.4	0.6

Packet loss in presence of a 4Mbps background traffic

In [5.13], X. Wang investigated the VoIP traffic volume. And he pointed out that IEEE 802.11b supports data rates up to 11Mbps and a typical VoIP stream requires less than 10Kbps. Ideally, the number of simultaneous VoIP streams that can be supported by an 802.11b WLAN is about 550 sessions. However, due to the added packet-header overheads

and the inefficiency of the WLAN MAC protocol, the experiments point out that the current WLAN can only support 12 simultaneous VoIP sessions. At the 802.11 MAC/PHY layers, the drop of efficiency is much worse. As a result, the overall efficiency drops to less than 3%. In an enterprise WLAN, the WLAN needs to support other complicated applications besides VoIP simultaneously. The TCP connections used by these applications can increase the delay and packet-loss rate of VoIP traffic dramatically. This decreases the quality of VoIP sessions.

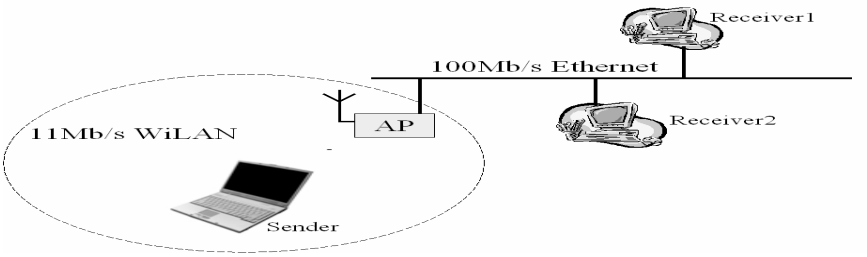


Figure 5.11: Network Scenario

In this work we propose a new digital watermarking-based integrity method for VoIP implementation over WLAN which is robust to considerable distortions often occurring in wireless networks. The scenario we are considering is shown in Figure 5.11. The network comprises a single IEEE 802.11 basic service set (BSS) with one access point (AP) and a number of mobile users. The AP is connected to a 100 Mb/s Ethernet, to which other users are directly connected. Voice calls take place between a user in the BSS and an user connected to the Ethernet (i.e., between the users Sender and Receiver1).

Chapter 6 Voice Cheque Proposal

6.1 Security of Mobile Commerce Applications

6.1.1 State of the Art

Mobile Commerce [6.1] is a hot topic driven by the fast and widely deployments of wireless networks and handheld devices. More and more customers prefer services like telephone banking or location-based tracking via their mobile phones or PDAs. Several successful operating environments are being widely used now, e.g., NTT DoMoCo's i-mode payment, Nordea's WAP Solo Mobile Banking Service, and Webraska's SmartZone Platform.

But if we look into their schemes carefully, we can find that their security depends heavily on the safety of removable/non-removable smart cards [6.2] issued by the telecommunication/mobile commerce providers. Even worse, although password is an easy and useful access control method, its disposure or misuse may cause disastrous results. In addition, with the fast growth of microelectronics technology, smart cards are not safe any longer [6.3]. Attackers can scan the smart cards, break the pin-codes, and fake them with other ones that hold the same information.

6.1.2 Biometric Identifiers

Any current Mobile Commerce scenario takes into account security schemes that in the end reside into "keeping a secret" (e.g., keeping always your SIM card in your possession). In the current work we suggest a solution that does not necessitate that. As we know human speech is a unique biometric identifier (as it is the DNA) and it does not change its characteristics. On the other hand, speech processing techniques [6.4] provide methods that help find the correct speaker of a piece of voice and they are so mature that even the distorted speech can be resolved very well. Our desire is to use this unique personal information (our special sound and way of speaking) when authenticating ourselves in order to obtain more (in terms of security) than when using smart cards. For this we address digital watermarking [6.5, 6.6], a multimedia processing technology developed in the past several years. By watermarking techniques we can insert unique digital watermarks into multimedia data to

protect its integrity. In our work, we integrate biometric identification and multimedia processing techniques to develop a new authentication approach that does not depend on the robustness of smart cards or passwords.

Using the technologies mentioned above, we propose a new approach, based on *Voice Cheques* (a concept to be introduced further on in the paper), to overcome three important security challenges of mobile commerce, naming:

- authentication of merchants and mobile customers;
- integrity of transaction data;
- non-repudiation of transaction.

The proposed approach is outlined in Figure 6.1. We assume two mobile phones being served by a Telecommunication Service Operator. A trustworthy third-party judger, the CA (Certification Authority), is in charge of generating unique digital certificates for each session and of distributing the digital certificate to the mobile phones. The mobile phones have a watermarking module that generates digital watermarks based on the received digital certificate and then watermark the outgoing voice stream as well as authenticate the incoming voice stream.

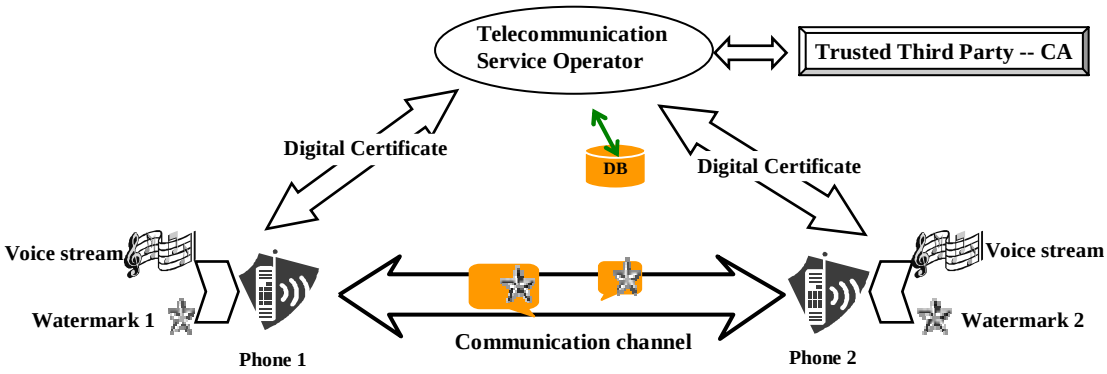


Figure 6.1: Voice Cheque Scheme

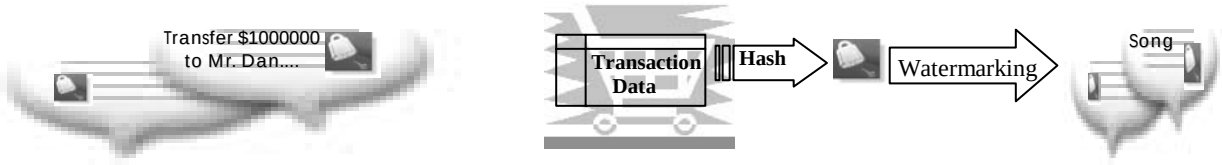
6.2 Definition of Voice Cheques

6.2.1 What is a Voice Cheque ?

We define a *Voice Cheque* as a human speech clip containing embedded digital watermarks. The embedded digital watermarks are generated based on the digital certificates issued by a Certificate Authority (trustworthy third-party). The digital watermark does not

degrade the quality of the speech clips and it is transparent to end users. The advantages of voice cheques are: providing very natural HMI to end users, generating legal proofs, non-repudiation of voice cheques, working smoothly in heterogeneous networks.

Telephone based M-Commerce services can be classified into two classes: voice based service and WAP based service. In our scenario and in the voice based banking service it is necessary for the staff of a bank and the customer to talk to each other on the phone for a short period of time. In this time the unique digital watermarks can be inserted into the voice streams to directly generate the voice cheques as shown in Figure 6.2(a). In a WAP based environment, at the end of each transaction, the customer needs as well to say a few words on the phone. We call this procedure "biometric identifier collection". A unique digital fingerprint is achieved by applying hash operation on the transaction data to identify this e-transaction. Then the digital watermarks are engraved into the collected voice to generate the voice cheque, as shown in Figure 6.2(b).



(a) Voice Cheque (b) WAP based Voice Cheque

Figure 6.2: Voice Cheque Scheme

6.2.2 General Scheme

In general, every cheque should have two properties: The first one is to state clearly who is the customer involved in the e-transaction; the second is whether or not this cheque has been modified (confidentiality). In our scheme, these issues have been addressed. With voice cheques, we can find the correct customer (the identification being based on his unique speech, the carrier of the voice cheque). In addition, the voice cheque is fragile. In other words, any modification on the transaction data will destroy the digital watermark embedded in voice cheque and so, the cheque itself.

In short, the propossed solution is similar to printing watermarks on the paper. The general scheme can be outlined in the following way:

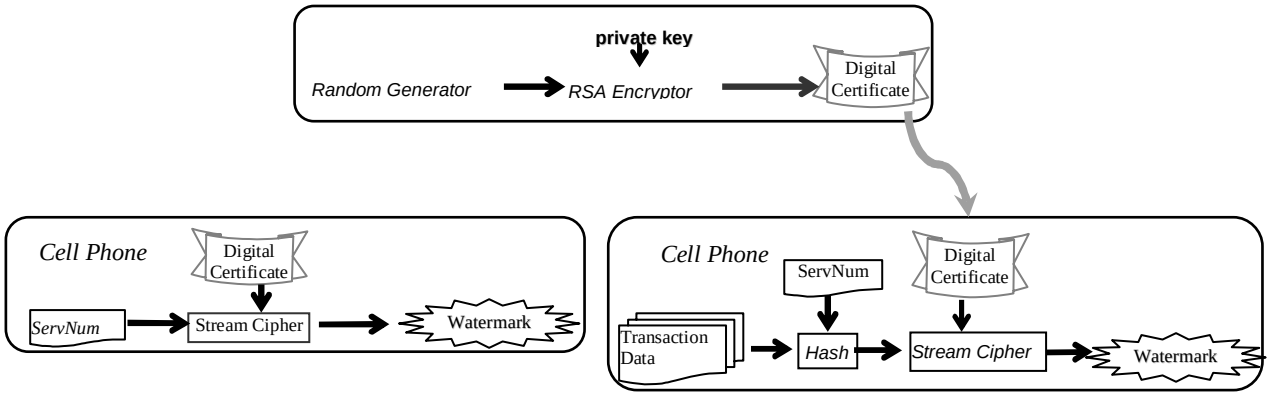
1) Collection of the end user's biometric identifier;
2) Generation of the digital watermarks;
3) Insertion of the digital watermarks into the biometric identifier, and in parallel, detection of the embedded digital watermarks from the watermarked multimedia data;

6.2.3 Digital Watermark Generation

In Figure 6.3 we illustrate the proposed digital watermark generation for voice based (Figure 6.3.a) and WAP based (Figure 6.3.b.) mobile phones. For the voice based M-Commerce models, after receiving the digital certificate, the identifier of this transaction, *ServNum*, is signed by the A8 stream cipher by using the digital certificate as the key to generate the digital watermark, as illustrated in Figure 6.3.a.. For the WAP based M-Commerce environments, the cell phone gets the digital fingerprint of the transaction data by applying a hash operation and the identifier of the transaction, *ServNum*, is the key to the hash facility. Then the intermediate result is signed by the A8 stream cipher using the digital certificate as the key to generate the digital watermark.

6.2.4 Digital Certificate Generation

Every session has an unique digital certificate which distinguishes the transaction from others. As shown in Figure 6.4, in the beginning of each transaction the CA creates one random number and then signs it using the RSA algorithm to generate a non-repudiant digital certificate.



(a) Voice Based Models (b) WAP Based Models

Figure 6.3: Digital Watermark Generation

6.2.5 Voice Cheque Generation

A cell phone samples the human speech from its microphone to create PCM raw voice and then compresses the raw voice data into a low-rate voice stream using the GSM speech coder. In parallel, the cell phone decompresses the incoming compressed voice stream.

To embed the digital watermarks into the voice stream or MMS-mail, the mobile phones should be slightly modified. A watermarking module needs to be installed on the mobile phone. Such a module consists of a watermark generator, a watermarking embedder and a watermarking extractor. Then the cell phone can take the task of digital watermark generation, embedding, and detection. Let us check them in detail. Once a cell phone receives the digital certificate from the CA, the watermark generator of the watermarking module creates immediately the watermark using the stream cipher; then the digital watermark is engraved into the human speech clips and, as well, synchronous signals are inserted into the speech clips to embed the period of voice clips. The watermarking module of the cell phone extracts the embedded digital watermarks from the incoming voice streams and compares them to the correct ones. If the incoming voice streams are acceptable, then the cell phones present them to the end users.

At this moment our system supports digital watermarking for GSM audio streams.

6.3 Authentication

6.3.1 Authentication Protocol

We integrate the authentication algorithms used in telecommunications [6.7, 6.8, 6.9, 6.10] with the mobile commerce authentication protocols.

A full session procedure occurring between two parties is a transaction. It includes the digital certificate handling and the voice stream transmission. Since many voice watermarking services may be taken at the same time, in order to make each transaction distinguishable, every transaction has a unique identifier, *ServNum*.

Similar to the commonly used authentication procedures, in this proposed authentication protocol each cell phone holds a shared key (k) with the CA. In the beginning of a mobile watermarking service, both cell phones send a *dialog_start* request to the CA and the *dialog_start* is encrypted using the stream cipher of the cell phones.

The CA authenticates the received *dialog_start* request and, if acceptable, generates a random number *RAND*, signs it using the RSA algorithm to create $RAND_{PK}$. The CA encrypts

$RAND_{PK}$ using the shared key, k, to construct the digital certificate and then sends the digital certificate to the cell phones and puts it on file, too.

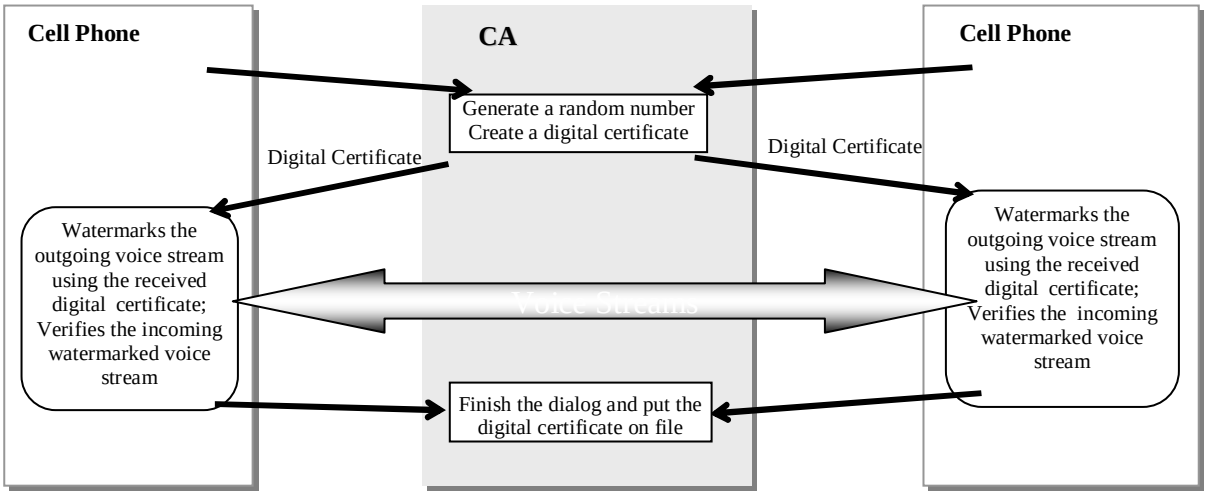


Figure 6.4: Outline of the Proposed Scheme

Phones----->CA { IMSI, TID, RQ_FLAG }$_k$
CA Authen{ IMSI, TID, RQ_FLAG }$_k$
Generate{ RAND }, Digital Certificate←Sign{ RAND}$_{PK}$
CA-------->Phones { Digital Certificate, TS, ServNum }$_k$
Phone Authen{ { DigCert, TS }$_k$ }$_k$
{
Two phones generate the digital watermarks, W, based on the Digital Certificate;
Digital watermarks are inserted into the voice stream;
Both phones authenticate the incoming voice stream.
}
Phones----->CA { TS, RQ_FLAG, ServNum }$_k$
CA Authen{ TS, Status_phone }$_k$
Put the digital certificate and ServNum into database

IMSI	International Mobile Subscriber Identity
TID	temporary ID
RQ	dialog request to CA
k	shared key between CA and a cell phone
{ X }k	encryption of a message, X, using k
{ X }PK	encryption of a message, X, using public key algorithm, key is PK
RAND	random number
TS	time stamp
ServNum	service number

Once the cell phone receives the digital certificate from the CA, the conversation between the two cell phones starts. At first, they authenticate the received digital certificate. If

the authentication is successful, then the cell phones generate the digital watermarks based on the received digital certificates. They then insert the digital watermarks into the users' voice streams. At the same time both of the phones extract the embedded digital watermarks from the incoming voice stream and authenticate the extracted digital watermarks.

At the end of the dialog each of the cell phone sends a *dialog_end* request to the CA. Finally, the CA authenticates the received *dialog_end* request and then puts them on file.
A notice worth to be mentioned here is that both of the participating phones do real-time authentication. This means that each of the mobile phones takes the tasks of inserting the watermarks into the voice stream and extracting the digital watermarks from the incoming voice streams in parallel.

6.3.2 Real-time Authentication

In order to ensure the overall safety of the communication, real-time authentication is employed during the transmission. As depicted in Figure 6.5, the Watermarking Embedding Module (Embedder) of one cell phone inserts the digital watermark into the outgoing voice stream and the digital certificate is the key. The Watermarking Extractor on the other cell phone produces the watermark from the incoming voice stream and uses the digital certificate issued by the CA as the key.

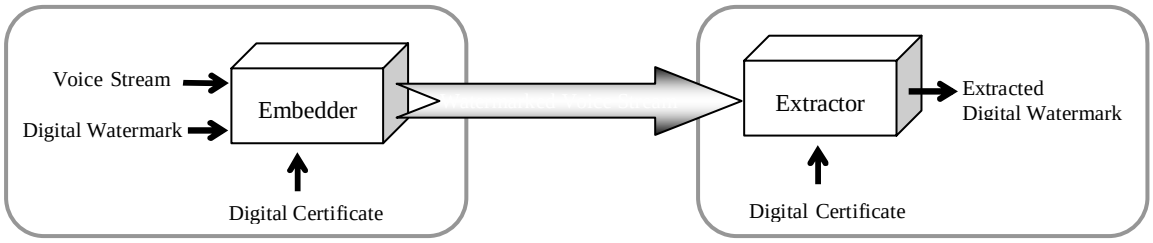


Figure 6.5: Digital Watermarking

6.3.3 Post Authentication

Post authentication is depicted in Figure 6.6.

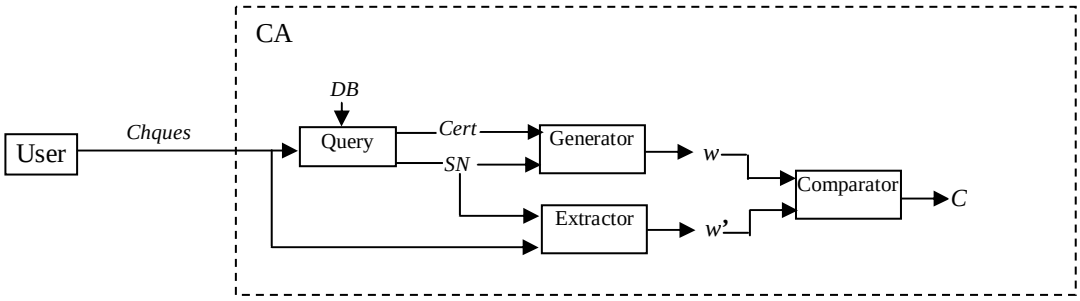


Figure 6.6 Digital Watermarking

The digital watermarks embedded in the voice clips are generated based on the digital certificates. These certificates have been signed by the CA by means of the RSA private key. This means that nobody, except the CA, can recreate the same digital watermarks at any time. Of course, each of the digital watermarks is different from the others.

The end users can record the conversation too. Once some of them come into any dispute, they can present the watermarked conversation to the CA. This authority then obtains the proper digital certificate, *Cert*, and the service number, *SN*, from the database, *DB*. It then re-generates the digital watermarks and extracts the digital watermarks from the speech clips. Finally, the CA compares it with the re-generated digital watermarks. If the extracted digital watermarks are the same as the re-generated watermarks, then the presented speech clips can be considered as proofs for a legal case.

1)	Users	Present the voice cheque to the CA (for WAP based voice cheque, the transaction data is also needed)
2)	CA	Search (Database) → Digital Certificate, ServNum
3)	CA	Authenticate (Digital Certificate, RSA's public key)
4)	CA	Generate (Digital Certificate, ServNum) → digital watermarks (for WAP based voice cheque, Hash (transaction data, ServNum) → digital fingerpint Generate (Digital Certificate, digital fingerprint)→ digital watermarks
5)	CA	Extract (voice cheque, ServNum) → extracted digital watermarks
6)	CA	Compare (generated digital watermarks, extracted digital watermarks)
7)	CA	Draw the conclusion

6.4. Experimental Results

A prototype has been implemented on desktop computers in order to verify the proposed approach.

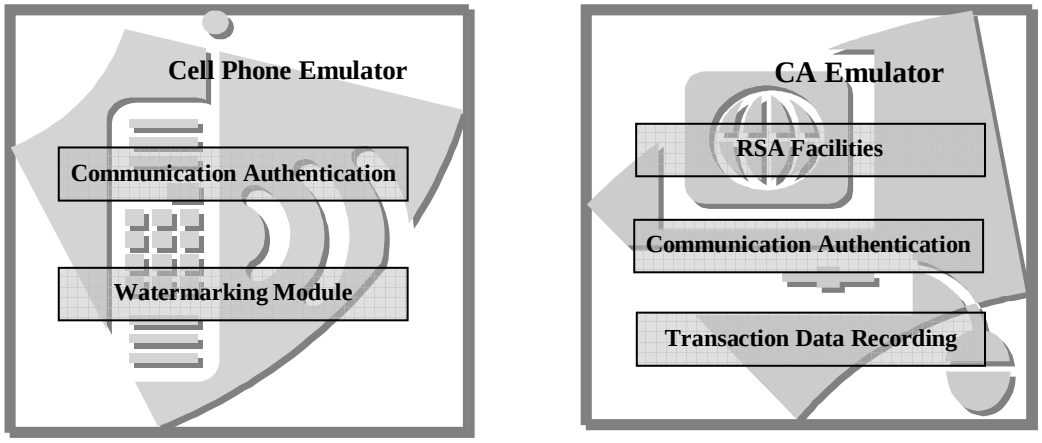


Figure 6.7 Prototype Architecture

We have used two cell phone emulators and a CA emulator. Each cell phone emulator used here contains a stream cipher and a watermarking module and the CA emulator consists of a random number generator, a RSA encryptor/decryptor and a stream cipher.

Chapter 7 Conclusions and Future Research

The work described in this thesis is concerned with the design of security system for multimedia communication using digital watermarking techniques. An overview of existing watermarking techniques was given and various applications of watermarks were introduced and. Digital watermarking was introduced as an alternative to conventional multimedia security methods. It overcomes the three basic drawbacks of traditional security mechanisms:

- First, the traditional methods are sensitive to data distortion — an unrecoverable bit error in the multimedia message or a packet drop during transmission may disable the corresponding authentication procedures.
- Secondly, the traditional methods create a checksum of the message and attach it to the target message. Hence, the checksum is apart from the message. A possible attacker may get access to the original multimedia message only and then reuse this message again and again to cheat the access control systems.
- Thirdly, these techniques increase the latency of multimedia communication considerably, so that the handheld devices (such as PDAs and cellular phones) in general cannot meet the resulting computing power requirements.

This thesis outlines a novel approach to extending well-known digital watermarking techniques to the protection of multimedia data during the network transmission within distributed highly asymmetric system architectures. An incremental watermarking algorithm suitable for low-resource embedded systems to be used in distributed environments is presented. The watermarked multimedia data authentication of the proposed scheme (which is the main contribution of this thesis) is public; its safety depends on the robustness of the underlying RSA algorithm. The main advantages of this approach were demonstrated for some use cases.

The first application scheme developed was based on digital image watermarking. This proposed scheme has several advantages and is suitable for real-time image collection: The architecture of the scheme is asymmetric; *WM_Server*'s computation task is heavy, while

WM_Sender's computation task is rather light. In addition, the image authentication is public, and a verification of public watermarks is also public.

Then, a security scheme for real-time speech communications using audio watermarking was proposed. The advocated approach adopts public key encryption to efficiently generate non-repudiate speech. Finally, a speech watermarking algorithm incorporating with GSM 610 full-rate coder was detailed.

The speech watermarking security architecture for VoIP has been talked by several researchers in last year.

While we mentioned specific future research directions at the end, we would also like to present a broad classification of intended future research directions for the sake of compactness. Our future research directions can be classified as follows:

(1) Implementation of the scheme presented in Chapter 4 in commercial telecommunication networks and furthers the potential e-commerce application. And the further investigation on the performance and usages should be taken. One of the usages of the proposed framework is a QoS measurement based on the impact of network on the multimedia data. This measurement will provide a more objective measurement on the performance of network traffic.

(2) Design of watermarking algorithms motivated by real-time video telephony: Video Telephony is of increasing interest nowadays. Video Telephony needs urgently to solve the privacy and confidentiality in order to exploit the M-Commerce applications. Nearly all of the mobile video telephony systems work on cell phones or pocket PC. Naturally, the next step is to find an answer to the question, "How hard is it to implement complex security solution on low-resourced mobile devices?" There are two basic approaches: multimedia data encryption and digital watermarking. Watermarking can be an easy way to provide the source origin authentication.

References

[1.1] I. Cox, M. Miller, J. Bloom, M. Miller, Digital Watermarking: Principles & Practice, *Morgan Kaufmann*, 2001

[1.2] M. Arnold, S. Wolthusen and M. Schmucker, Techniques and Applications of Digital Watermarking and Content Protection, *Artech House Publishers*, 2003

[1.2] Jiang J. and A. Armstrong 'A data hiding approach for efficient image indexing' *IEE Electronic Letters*, 38 (23), 2002

[1.3] A. Bhardwaj, T. P. Pandey and S. Gupta, Joint Indexing and Watermarking of Video Using Colour Information, *Workshop on Multimedia Signal Processing*, Cannes, France, 2001

[1.4] Webster College Dictionary, 1998

[2.1] M. Baugher et al., The Secure Real Time Transport Protocol, *IETF Draft*, 2002

[2.2] N. Doraswamy and D. Harkins, IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, *Prentice Hall*, 1999

[2.3] I. Agi and L. Gong, An Empirical Study of MPEG Video Transmissions, *Proceedings of The Internet Society Symposium on Network and Distributed System Security*, pp.137-144, 1996

[2.4] L. Tang, Methods for Encrypting and Decrypting MPEG Video Data Efficiently, *ACM Multimedia*, 1996

[2.5] L. Qiao and K. Nahrstedt, A New Algorithm for Video Encryption, *Proceedings of the First International Conference on Imaging Science, Systems and Technology*, 1997

[2.6] IETF RFC 8042, June 2005

[2.7] IETF RFC 2402, November 1998

[2.8] C. Perkins, RTP: Audio and Video for the Internet, *Addison Wesley*, June 2003

[2.9] I. Cox, M. Miller and A. McKellips, Watermarking as Communications with Side Information, *Proceedings of the IEEE, 87, 7, 1127-1141*, 1999

[2.10] IETF H.235,

[2.11] RTP Specification, IETF RFC 1889, 1996.

[2.12] DES-SBC, RFC 1423, *IETF*, 1993

[2.13] S. Bosworth, A. Hutt and D. Hoyt, Computer Security Handbook, *John Wiley & Sons*, 2002

[2.14] www.pgpi.org

[3.1] C.I., Podilchuk, E.Delp, Digital Watermarking: Algorithms and Applications, *IEEE signal Processing Magazine*, 18/4, 2001.

[3.2] L. Xie and G. Arce, Joint Wavelet Compression and Authentication Watermarking, *Proceedings of the IEEE International Conference on Image Processing*, ICIP '98, Chicago, 1998

[3.3] I.J. Cox, J. Kilian, T. Leighton and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *IEEE Trans. on Image Processing*, 6/12, 1997

[3.4] F. Hartung and M. Kutter, Multimedia Watermarking Techniques, *Proceedings of the IEEE*, 87/7, 1999

[3.5] M. Scheneider and S. Chang, A Robust Content Based Digital Signature for Image Authentications, *Proceedings of IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, Atlanta, 1996

[3.6] F.Hartung and M.Kutter, Multimedia Watermarking Techniques, *IEEE Proc.*, Vol. 87, No. 7, pp.1079-1107, July 1999

[3.7] F. Peticolas, R. Anderson, and M. Kuhn. Information Hiding - a Survey, *IEEE Proc.*, Vol. 87, No.7, pp. 1062-1078, July 1999

[3.8] H. Huang, F. Wang, and J. Pan, Efficient and Robust Watermarking Algorithm with Vector Quantisation. *Electron. Letter*, June 2001

[3.9] T. Kalker and J. Haitsma, Efficient Detection of a Spatial Spread Spectrum Watermark in MPEG Video Streams. *Proceeding of IEEE Int. Conf. Image Processing*, 2000

[3.10] C. Lu, S. Huang, C. Sze, and H. Liao, Cocktail Watermarking for Digital Image Protection. *IEEE Transaction on Multimedia*,,4 , 2000

[3.11] P. Barreto, H. Kim, and V. Rijmen, Toward a Secure Public-Key Blockwise Fragile Authentication Watermarking. *Proceeding of IEEE Int. Conf. Image Processing*, 2001

[3.12] M. Yeung and F. Minzter, An Invisible Watermarking Technique for Image Verification, *Proceeding of IEEE Int. Conf. Image Process*, pp.680-683, 1997

[3.13] C. Li and F. Yang. One-Dimensional Neighbourhood Forming Strategy for Fragile Watermarking, *Journal of Electronic Imaging*, 12/ 2, 2003

[3.14] M. Wu and B.Liu. Watermarking for Image Authentication, *Proceeding of IEEE Int. Conf. Image Processing*, II, pp.437-441, October 1998

[3.15] L. Xie and G. Arce. A Class of Authentication Digital Watermarks for Secure Mmultimedia Communication. *IEEE Transaction on Image Processing*, November 2001

[3.16] H. Inoue, A. Miyazaki, and T. Katsure. Wavelet-Based Watermarking for Tamper Proofing of Still Images, *Proceeding of IEEE Int. Conf. Image Processing*, September 2000

[3.17] X. Zhou, X. Duan, and D. Wang. A Semi-Fragile Watermark Scheme For Image Authentication. *10th IEEE Int. Multimedia Modelling Conf.* January 2004
[3.18] Stephane Mallat, A Wavelet Tour of Signal Processing, *Academic Press*, 1999
[3.19] www.mathwork.com
[3.20] K. Rao and J. Hwang, Techniques & Standards for Image Video & Audio Coding, *Prentice Hall*, 1996
[3.21] B. Chen and G. Wornell, Dither Modulation: A New Approach to Digital Datermarking and Information Embedding, *Proceedings of theSPIE: Security and Watermarking of Multimedia Contents*, vol. 3657, 1996
[3.22] P. Goupillaud, A. Grossman and J. Morlet. Cycle-Octave and Related Transforms in Seismic Signal Analysis, *Geoexploration*, Vol. 23, 1984.
[4.1] W. Bender and D. Gruhl, Techniques for Data Hiding, *IBM Systems Journal*, 35/3, pp.313-336, 1996
[4.2] J. Fridrich, M. Goljan and R. Du, Distortion-Free Data Embedding, *Lecture Notes in Computer Science 2173*, pp. 27–41, 2001
[4.3] Y. Lee and L. Chen, High Capacity Image Steganographic Model, *IEE Proceedings Vision Image Signal Processing 147/3*, pp. 288–294, 2000
[4.4] J. Fridrich, M. Goljan and R. Du, Lossless Data Embedding - New Paradigm in Digital Watermarking, *Applied Signal Processing 2002/2*, pp. 185–196, 2002
[4.5] C. Yeh and C. Kuo, Digital Watermarking Through Quasi M-Arrays, *Proceedings of IEEE Workshop on Signal Processing Systems*, pp. 456–461, 1999
[4.6] T. Ciloglu and S. Karaaslan, An Improved All-Pass Watermarking Scheme for Speech and Audio, *Proeeding of IEEE International Conference on Multimedia and Expo*, pp. 1017–1020, 2000
[4.7] R. Lancini, F. Mapelli and S. Tubaro, Embedding Indexing Information in Audio Signal Using Watermarking Technique, *Proceeding of 4th EURASIP-IEEE International Symposium on Video/Image Processing and Multimedia Communications*, pp. 257–261, 2002
[4.8] D. Huang and T. Yeo, Robust and Inaudible Multi-Echo Audio Watermarking, *Proceeding of IEEE Pacific-Rim Conference On Multimedia*, pp. 615–622, 2002
[4.9] B. Ko, R. Nishimura and Y. Suzuki, Time-Spread Echo Method for Digital Audio Watermarking Using PN Sequences, *Proceeding of. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2001–2004, 2002
[4.10] I. Cox, J. Kilian, F. Leighton and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *IEEE Transactions on Image Processing*, Dec 1997

[4.11] D. Kirovski and H. Malvar, Spread-Spectrum Watermarking of Audio Signals, *IEEE Transactions on Signal Processing*, Apr 2003
[4.12] C. Xu , J. Wu and Q. Sun, A Robust Digital Audio Watermarking Technique, *Proceeding of International Symposium on Signal Processing and its Applications*, Australia, 1999
[4.13] A. Lemma, J. Aprea, W. Oomen and L. Kerkhof, A Temporal Domain Audio Watermarking Technique, *IEEE Transactions on Signal Processing*, Apr 2003
[4.14] GSM Vocoder Specification, *ITU*, 1996
[4.15] P. Nguyen, A Study on Efficient Algorithms for Temporal Decomposition of Speech, *PhD thesis*, JIAST, 2004.
[4.16] J. Herre and J.D. Johnston, Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS), *Proceeding of 101st AES Convention*, Nov 1996.
[4.17] M. Athineos and D. Ellis, Frequency Domain Linear Prediction for Temporal Features, *Proceeding of 101st AES Convention*, Nov 1996.
[4.18] www.avs.org.cn, 2002.
[4.19] H. Hermansky, Perceptual Linear Predictive Analysis of Speech, *Journal of the Acoustical Society of America*, Vol. 87:1738-1752, 1990.
[4.20] T. Li, Computerized Sound, *Science Press*, Beijing, 2000.
[4.21] J. Dai, Linear Algebra, *Northeas Press*, Beijing, 2000.
[4.22] Z. Liu, Y. Kobayashi, S. Sawato and A. Inoue, A Robust Audio Watermarking Method Using Sine Function Patterns Based on Pseudo-Random Sequences, *Proceedings of Pacific Rim Workshop on Digital Steganography*, 2002.
[4.23] R. McAulay and T. Quatieri, Speech Analysis/Synthesis Based on a Sinusoidal Representation, IEEE Transactions on Acoustics, Speech and Signal Processing Vol. 34, No. 4, 1986
[4.24] X. Serra and J. Smith, Spectral Modelling Synthesis : a Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition, *Computer Music Journal*, Vol. 14, No. 4, 1990.
[5.1] IEEE Draft International Standards, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, *IEEE P801.11/D10*, 1999.
[5.2]http://www.tmcnet.com/articles/itmag/0199/0199roundt.htm.
[5.3] M. Baugher et al., The Secure Real Time Transport Protocol, *IETF Draft*, 2002.
[5.4] Z. Li, R. Xu, Energy Impact of Secure Computation on a Handheld Device, *IEEE International Workshop on Workload Characterization*, pp.109 –117, 2002.

[5.5] C. Wu and C. Kuo. Speech Content Integrity Verification Integrated with ITU g.723.1 Speech Coding, *IEEE International Conference on Information Technology: Coding and Computing*, pp. 680–684, 2001.
[5.6] J. Haitsma, M. Veen, T. Kalker and F. Bruekers, Audio Watermarking for Monitoring and Copy Protection, *Proceedings of the 2000 ACM workshops on Multimedia*, pp.119–122, 2000.
[5.7] D. Gruhl, W. Bender, and A. Lu, Echo-hiding, *Information Hiding: 1st International Workshop*, pp.295–315, 1996.
[5.8] Brian C. Wiles and John Walker, *Speakfreely*.
[5.9] http://www.openssl.org.
[5.10] C. Perkins, O. Hodson, and V. Hardman, A Survey of Packet Loss Recovery Techniques for Streaming Media, *IEEE Network Magazine*, September/October 1998.
[5.11] C. Xu, J. Wu and Q. Sun, Digital Audio Watermarking and its Application in Multimedia Database, *Fifth International Symposium on Signal Processing and Its Applications*, 1999
[5.12] Megaco, RFC 3015, IETF, 2003
[5.13] X. Wang, G. Min, and J. Mellor, Improving VoIP Application's Performance over WLAN Using a New Distributed Fair MAC Scheme, *IEEE AINA*, March 2004
[5.14] SIP, RFC 2543, IETF, 1999
[5.15] MGCP, RFC 3435, IETF, 2003
[6.1] Norman Sadeh, M-Commerce, *Wiley Computer Publishing*, 2002
[6.2] R. Sanchez-Reillo, Achieving security in Integrated Circuit Card applications: reality or desire?, Security Technology, 2001 IEEE 35th International Carnahan Conference on , Oct 2001
[6.3] http://archive.infoworld.com/articles/hn/xml/02/05/14/020514hncambridge.xml
[6.4] P. Woodland, Speech Recognition, *Speech and Language Engineering - State of the Art (Ref. No. 1998/499), IEE Colloquium on*, Nov 1998
[6.5] C.I., Podilchuk, E. Delp, Digital Watermarking: Algorithms and Applications", *IEEE Signal Processing Magazine*, Vol. 18, Issue 4, 2001
[6.6] I.J. Cox, J. Kilian, T. Leighton and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *IEEE Trans. on Image Processing*, Vol. 6, Issue 12, 1997
[6.7] J.C, Cooke, R.L. Brewster, Cryptographic Algorithms and Protocols for Personal Communication Systems Security, *IEE Colloquium on Security and Cryptography Applications to Radio Systems*, 1994

[6.8] H.Y.Lin and L. Harn, Authentication Protocols for Personal Communication Systems", *Proceedings of ACM SIGCOMM'95*, August 1995

[6.9] N. Asokan, Anonymity in a Mobile Computing Environment, *Proceedings of Workshop on Mobile Computing Systems and Applications*, pp. 200-204, 1994

[6.10] D. Samfat, R. Molva and N. Asokan, "Untraceability in mobile networks",*Proceedings of Int. Conf. on Mobile Computing and Networking*, pp. 26-36, 1995

Curriculum Vitae

Name: Yuan, Song

Data of Birth: August 21, 1975

Education:

2000 Master, CAS, China

1997 Bachelor, Northeast University, China

Work Experiments:

2002-2004 Research Assistant, ISS, TU Darmstadt

2000-2001 Software Engineer, Sail-Lab GmbH