In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

Multiple Packet-Streams in Encrypted Voice over IP

submitted by

Shuman Jin, B. Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering
Faculty of Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario, Canada K1S 5B6
May 2003

© COPYRIGHT 2003, SHUMAN JIN

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88877-0

Our file Notre référence

ISBN: 0-612-88877-0

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

ABSTRACT

Voice over Internet Protocol (VoIP) is the technology that enables voice packets transmission over packet switched networks. Security is very important due to the popularity of VoIP.

The nature of IP networks brings packet latency, delay jitter and packet loss. These issues must be taken into account where cryptographic systems are implemented. The impact of a cryptographic system should be minimized so as to provide acceptable quality of service. When Ciphertext Block Chaining (CBC) or Ciphertext Feedback (CFB) modes are chosen for the popular DES encryption algorithm, the packet loss due to latency propagates to next packet, thus increasing packet loss and degrading quality of service.

In this thesis, we propose a multiple packet-stream scheme for the encrypted VoIP system with CBC mode or CFB mode. Our simulation results show the proposed scheme not only strengthens security by using multiple secret keys but also reduces packet loss. Real speech recording and artificial speech are used, VAD and frame interleaving schemes are also implemented.

ACKNOWLEDGEMENTS

I would like to acknowledge my supervisors, Dr. Rafik A. Goubran and Dr. Richard M. Dansereau for their professional and heuristic guidance and support of my research for this thesis.

I also would like to thank all the students in the DSP lab for their valuable information and help.

I am grateful to all the technicians in the Department of Systems and Computer Engineering for providing a proper research environment for me to accomplish my work.

I would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada and Communications and Information Technology Ontario (CITO) for funding.

Finally, I would like to thank my family for their understanding and support so that I could concentrate on my research.

TABLE OF CONTENTS

ABSTRACT.....	III
ACKNOWLEDGEMENTS.....	IV
TABLE OF CONTENTS.....	V
LIST OF TABLES.....	VIII
LIST OF FIGURES.....	IX
LIST OF ABBREVIATIONS.....	XII

Chapter 1	Introduction	1
1.1	Overview	1
1.1.1	<i>Voice over IP</i>	1
1.1.2	<i>Security of VoIP</i>	2
1.2	Problem Statement.....	3
1.3	Goals of the Thesis	5
1.4	Summary.....	5
1.5	Thesis Outline.....	5
Chapter 2	Background Review.....	2
2.1	Voice over IP	2
2.1.1	<i>VoIP Related Protocols and Standards</i>	2
2.1.2	<i>Voice Coding</i>	10
2.1.3	<i>Characteristics of VoIP</i>	10
2.1.4	<i>Voice Activity Detection (VAD)</i>	16
2.1.5	<i>Error Control and Concealment Schemes</i>	19

2.2 Cryptography and Cryptographic Systems	21
2.2.1 Basic Cryptographic Concepts and Requirements	21
2.2.2 Cryptosystem Implementation Options	23
2.2.3 Stream Cipher and Block Cipher	24
2.2.4 Types of Cryptosystems	24
2.2.5 DES Modes of Operation	31
2.2.6 Comparisons of Secret Key Cryptosystem and Public Key Cryptosystem	41
2.2.7 Issues of Secure VoIP	42
Chapter 3 The Multiple Packet-Stream Scheme	43
3.1 Encrypted Packet-Stream Scheme	43
3.2 Benefits of Multiple Packet-Stream Scheme	46
3.2.1 Reduced Error Propagation Effects	46
3.2.2 Enhanced Message Security	48
Chapter 4 Simulation Setup	50
4.1 The Simulation Structure	50
4.2 Voice Packet Considerations	53
4.3 Network Delay	57
4.3.1 Long-term Delay	57
4.3.2 Short-term Delay	58
4.4 Packet Delay	61
4.5 Buffer Size and Play Out Schedule	62
Chapter 5 Simulation Results Analysis	63
5.1 Effect of Number of Packet-Streams	63
5.1.1 Simulation Results For Real Speech	63
5.1.2 Simulation Results for Artificial Speech	69
5.1.3 Overall Effect of Multiple Packet-Stream Scheme	73
5.2 Effect of Play Out Delay	75
5.3 Effect of Decryption Buffer and Jitter Buffer	79
5.3.1 Effect of Decryption Buffer	79
5.3.2 Effect of Jitter Buffer	82
5.4 Effect of Frame-Interleaving	88

Chapter 6	Conclusions.....	91
Appendix A:	Additional Simulation Results Analysis.....	94
REFERENCES.....		100

LIST OF TABLES

Table 2. 1 End-to-end VoIP Packet Delays	13
Table 2. 2 Diffie-Hellman Algorithm	30
Table 3. 1 Effect of Multiple Packet-Streams on Protecting Message Privacy	49
Table 4. 1 Typical Distribution Parameters of Artificial Speech.....	56
Table 5. 1 Simulation Results for Real Speeches	67
Table 5. 2 Simulation Results for Three Types of Artificial Speeches	72
Table 5. 3 Required Jitter Buffer Size for Minimum Packet Loss.....	87

LIST OF FIGURES

Figure 1.1 Structure of Encrypted VoIP System	3
Figure 2. 1 Format of Voice Packet.....	11
Figure 2. 2 Packet Inter-arrival Time and Play Out Scheme Illustration.....	15
Figure 2. 3 Waveform of A Sample Speech Recording.....	18
Figure 2. 4 Frame Interleaving Scheme.....	20
Figure 2. 5 DES Encryption Computation.....	26
Figure 2. 6 TDEA Encryption/Decryption Operations	28
Figure 2. 7 The Electronic Codebook Mode.....	32
Figure 2. 8 The Cipher Block Chaining Mode	34
Figure 2. 9 Error Propagation of the CBC Mode.....	34
Figure 2. 10 The Cipher Feedback Mode	36
Figure 2. 11 Error Propagation of the CFB Mode	37
Figure 2. 12 The Output Feedback Mode	38
Figure 2. 13 The Counter Mode.....	40
Figure 3. 1 The Structure of Multiple Packet-Stream Scheme	44
Figure 3. 2 Effect of Multiple Packet-Stream Scheme on Packet Loss	47
Figure 4. 1 The Buffer Queuing Structure.....	51
Figure 4. 2 Flowchart of Buffer Receiving/Dropping Algorithm.....	52
Figure 4. 3 Packet Storage in Buffers with n Multiple Packet-Streams	53
Figure 4. 4 Energy-based VAD Results ($k=100$).....	55
Figure 4. 5 The On-Off Patterns of Artificial Speeches	57
Figure 4. 6 Typical Mean Delays in a Weekday	58
Figure 4. 7 Pareto Distribution Delays	60
Figure 4. 8 Pareto Probability Distribution Function.....	63
Figure 5. 1 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech#1)	65
Figure 5. 2 Effect of Multiple Packet-Streams on Length of Loss Segment(real speech#1).....	66
Figure 5. 3 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech#2)	66

Figure 5. 4 Effect of Multiple Packet-Streams on Length of Loss Segment(real speech#2).....	67
Figure 5. 5 Effect of Multiple Packet-Streams on Packet Loss rate (type-1)	69
Figure 5. 6 Effect of Multiple Packet-Streams on Length of Loss Segment (type-1)	70
Figure 5. 7 Effect of Multiple Packet-Streams on Packet Loss Rate (type-2).....	70
Figure 5. 8 Effect of Multiple Packet-Streams on Length of Loss Segment (type-2)	71
Figure 5. 9 Effect of Multiple Packet-Streams on Packet Loss Rate (type-3).....	71
Figure 5. 10 Effect of Multiple Packet-Streams on Length of Loss Segment (type-3) ..	72
Figure 5. 11 Overall Effect of Multiple Packet-Streams on Packet Loss Rate.....	74
Figure 5. 12 Overall Effect of Multiple Packet-Streams on Length of Loss Segment ...	74
Figure 5. 13 Effect on Packet Loss Rate for Different Play Out Delays (real speech#1)	77
Figure 5. 14 Effect on Length of Loss for Different Play Out Delays (real speech#1) ..	77
Figure 5. 15 Effect on Packet Loss Rate for Different Play Out Delays (type-1)	78
Figure 5. 16 Effect on Length of Loss for Different Play out Delays (type-1).....	78
Figure 5. 17 Effect of Decryption Buffer on Packet Loss Rate (real speech#1)	79
Figure 5. 18 Effect of Decryption Buffer on Length of Loss Segment (real speech#1) .	80
Figure 5. 19 Effect of Decryption Buffer on Packet Loss Rate (type-1)	80
Figure 5. 20 Effect of Decryption Buffer on Length of Loss Segment (type-1)	81
Figure 5. 21 Effect of Jitter Buffer on Packet Loss Rate (real speech#1, DB=3).....	83
Figure 5. 22 Effect of Jitter Buffer on Packet Loss Rate (real speech#1, DB=5).....	83
Figure 5. 23 Effect of Jitter Buffer on Length of Loss Segment (real speech#1, DB=3)	84
Figure 5. 24 Effect of Jitter Buffer on Length of Loss Segment (real speech #1, DB=5)	84
Figure 5. 25 Effect of Jitter Buffer on Packet Loss Rate (type-1, DB=3).....	85
Figure 5. 26 Effect of Jitter Buffer on Packet Loss Rate (type-1, DB=5).....	85
Figure 5. 27 Effect of Jitter Buffer on Length of Loss Segment (type-1, DB=3).....	86
Figure 5. 28 Effect of Jitter Buffer on Length of Loss Segment (type-1, DB=5).....	86
Figure 5. 29 Effect on Packet Loss Rate with Frame-Interleaving Implementation	90
Figure 5. 30 Effect on Length of Loss Segment with Frame-Interleaving implementation	90

Figure A. 1 Delays Generated by A Bursty Network Delay Model	94
Figure A. 2 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech #1) ..	95
Figure A. 3 Effect of Multiple Packet-Streams on Length of Loss Segment (real speech#1).....	97
Figure A. 4 Effect of Multiple Packet-Streams on Packet Loss Rate (type-1)	97
Figure A. 5 Effect of Multiple Packet-Streams on Length of Loss Segment (type-1) ...	98
Figure A. 6 Effect of Multiple Packet-Streams on Packet Loss Rate (type-2)	98
Figure A. 7 Effect of Multiple Packet-Streams on Length of Loss Segment (type-2) ...	99

LIST OF ABBREVIATIONS

A

AES Advanced Encryption Standard

C

CBC Cipher Block Chaining mode

CFB Cipher Feedback mode

CNG Comfort Noise Generation

CTR Counter mode

D

DES Data Encryption Standard

DH Diffie-Hellman algorithm

E

ECB Electronic Codebook mode

F

FEC Forward Error Control

FIPS PUB Federal Information Processing Standards Publication

I

IETF Internet Engineering Task Force

IP Internet Protocol

ITU International Telecommunication Union

IV Initialization Vector

L

LAN Local Area Network

N

NIST National Institute of Standards and Technology

NTP Network Time Protocol

O

OFB Output Feedback mode

P

PCM Pulse Code Modulation

PDF Probability Distribution Function

PSTN Public Switched Telephone Protocol

Q

QoS Quality of Service

R

RFC Request For Comments

RSA Ron Rivest, Adi Shamir, and Leonard Adleman algorithm

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

S

SIP Session Initiation Protocol

T

TCP Transmission Control Protocol

TDEA Triple Data Encryption Algorithm

U

UDP User Datagram Protocol

V

VAD Voice Activity Detection

VoIP Voice over Internet Protocol

W

WAN Wide Area Network

Chapter 1 Introduction

1.1 Overview

The prevalence of various multimedia applications over Internet Protocols (IP) brought us significant convenience and efficiency in information exchange. Whenever the communication is internal or carried on a private network, people aren't concern too much about security. However, when protected data transmissions are required over open and unsecured communication networks, cryptographic systems are desirable to maintain secrecy.

1.1.1 Voice over IP

The transmission of voice over a network is not a new technology. VoIP allows integration of voice, data and multimedia applications on a single network. In comparison the traditional Public Switched Telephone Protocol (PSTN) can provide adequate Quality of Service (QoS) or toll quality, but is very inefficient in terms of bandwidth utilization. Because a conversation usually has many silence intervals, the VoIP technology can suppress these intervals without sending them to the network, and then reconstruct them at the receiving end. The present Internet network was initially designed for data transmissions but not for interactive applications, it can only provide best-effort communications and offer unguaranteed QoS. UDP voice packets transmitted over IP may be routed through different paths, thus packets may experience varying queuing delays and suffer from packet losses [1][2]. These impairments are very important

issues in VoIP. Many researches have been done to deal with these problems in time-sensitive applications. For example, resources can be reserved for interactive application and higher priority can be assigned to real-time audio and voice packets over data packets; jitter buffers are often used to smooth delay jitter and dropped late packets are made up to obtain better QoS.

1.1.2 Security of VoIP

Although wiretapping is one of the concerns in PSTN, people usually consider the security provided by PSTN is reasonable and sufficient. If an enemy is trying to eavesdrop on a conversation over PSTN, he must be physically connected to the telephone line and have access to certain hardware devices. And only one conversation is tapped at a time. But in a VoIP scenario, the threats of such kind of attack increase considerably. Because IP networks are used by all regular users, any VoIP conversation transmitted over the IP networks may be compromised by anybody who has regular access to the networks. Although certain equipment and tools are needed, they are not out of reach of today's expert hackers. Data sniffing tools are readily available and it will not take too long for these tools to be aware of new VoIP protocols [3]. VoIP traffic can be sniffed, saved and played back at a later time for attack. Encrypted data may be intercepted and broken offline without being even noticed. For all these reasons, information security is of great importance as more and more companies use IP networks to exchange sensitive information.

It is well known that encryption and decryption algorithms usually involve extensive computation. While applying cryptographic systems (cryptosystems) to VoIP applications with the security concerns in mind, one of the imperative tasks is to

minimize the impact on QoS and the performance of VoIP within the stringent real-time constraints. Although in the past few years VoIP has developed rapidly and is becoming a mature technology, there is still a lot of work to do to protect conversations delivered over the Internet.

Figure 1.1 below shows a general structure of cryptographic VoIP system.

Figure 1.1 Structure of Encrypted VoIP System

1.2 Problem Statement

Packetized voice transmitted over IP often encounters varying delays and delay jitter and packet loss. Delay is due to varying delivery paths and queuing delays, delay jitter is the variance of packet interarrival time and packet loss is usually due to delay. To smooth the delay jitter, packets are buffered and then replayed with a play out delay. Late packets are considered packet losses. Because VoIP is a time critical application, certain play out delay and packet loss requirements must be met so as to produce acceptable

voice quality. Whenever a cryptographic system is implemented in VoIP, those requirements must be taken into account.

Data Encryption Standard (DES), specified by the National Institute of Standards and Technology (NIST), is a widely implemented cryptographic system. Among the operation modes offered by DES, those modes that chain data with feedback in encryption such as Cipher Block Chaining Mode (CBC) or Cipher FeedBack Mode (CFB) are usually selected because of their stronger security. These encryption modes use previous encrypted packets as feedback for subsequent packet encryption and use the same feedback for subsequent encrypted packet decryption. Whenever an encrypted packet is lost or in error (erroneous, missing or gaining bits in the needed packet), the decryption of the next packet is erroneous. This is called error propagation. In other words, the packet decryption depends on the availability of its previous encrypted packet at the receiving end.

The late encrypted voice packets that arrive after a certain time are not only dropped and considered losses but also cause decryption failure of successive packets when CBC mode and CFB mode are used. Therefore the packet loss increases and consequently degrades the performance of VoIP. In order to reduce the effect of error propagation, the time constraint of late packets for subsequent packets decryption should be relaxed so that the subsequent packets can still make use of late packets for decryption. In our investigation, we only consider error propagation produced by late packets – packets are encrypted, decrypted and transmitted without error and considered correct as long as they arrive in time for replay or decryption.

1.3 Goals of the Thesis

We propose a multiple packet-stream scheme using a jitter buffer and a decryption buffer so as to reduce error propagation and strengthen security. The objectives of this thesis are to investigate the following in an encrypted Voice over IP scenario.

- Effect of the number of packet-streams on packet loss
- Effect on maintaining message privacy
- Effect of the play out delay on the multiple packet-stream scheme
- Effect of the jitter buffer and decryption buffer on the multiple packet-stream scheme
- The effect of the frame-interleaving scheme on the multiple packet-stream scheme

1.4 Summary

The main contribution of the thesis is the multiple packet-stream scheme. It can significantly relax the time-constraint of voice packets for decryption purposes, thus reducing error propagation and packet losses, and improving QoS. In this scheme, each packet-stream can use a different encryption key so as to strengthen security without introducing much extra resource consumption and delay. It is also flexible and cost efficient because it can be easily implemented in software.

1.5 Thesis Outline

The rest of the thesis is structured as follows: Chapter 2 provides an overview of the Voice over IP technology and introduces the basic concepts of cryptography and popular cryptographic algorithms. Chapter 3 presents our multiple packet-stream proposal and addresses its benefits. Chapter 4 describes the simulation environment

considerations and setup and chapter 5 analyzes and discusses the simulation results. Finally, conclusions are provided in chapter 6. More simulation results analysis are included in the Appendix A.

Chapter 2 Background Review

2.1 Voice over IP

Voice over IP is the technology that digitizes voice signals, packetizes voice data, delivers voice packets over Internet Protocol networks and converts digital data into voice signals. The increased bandwidth, improved compression technology and convenient and easy access to the Internet make VoIP a feasible technology.

VoIP is a challenging application for QoS. Transmission of voice traffic has stringent requirements on latency, delay jitter and packet loss. Unlike traditional PSTN, which transmits voice traffic over a dedicated circuit, VoIP delivers voice packets in a best-effort manner. Packets may be transmitted over arbitrary routes and have no fixed delay even if packets are generated and transmitted with constant time interval at the source end.

2.1.1 VoIP Related Protocols and Standards

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are transport layer protocols of IP. Real-time Transport Protocol (RTP) and Real-time Transport Control Protocol (RTCP) are protocols for real-time applications. H.323 is a standard for videoconferencing and Session Initiation Protocol (SIP) is a signaling protocol for VoIP. H.235 is a security and encryption protocol for H-Series multimedia terminals.

1. Transmission Control Protocol (TCP)

TCP [4] is a connection-oriented protocol that provides reliable end-to-end transport service by using a retransmission scheme. Although TCP can retransmit a packet until it is correctly received and there is no need to re-order packets, it may lead to a long delay, which is not acceptable for a real-time service such as VoIP. So TCP is not suitable for voice message delivery.

2. *User Datagram Protocol (UDP)*

UDP [5] is a connectionless-oriented protocol providing an unreliable transport service wherein packet delivery is not guaranteed. Each packet may go through a different path and encounter varying delay. Because packet arrivals are not necessarily in sequence, re-ordering is usually needed. UDP can produce shorter delay without packet retransmission, but at the expense of higher packet loss rate and the need of re-ordering. Since voice packets usually contain redundant information, a certain level of packet loss rate is better tolerated and less critical in VoIP, compared to a long delay.

3. *Real-time Transport Protocol (RTP)*

RTP [6] provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality. It can monitor data transportation by using sequence number, time stamp and data type identifiers.

4. *Real-time Transport Control Protocol (RTCP)*

RTCP is a protocol that participates with RTP and fulfills the task of performance and behavior monitoring [6]. It is based on the periodic transmission of control packets to

all participants in the session.

5. H.323

International Telecommunication Union (ITU) H.323 [7] is a set of protocols for voice, video and data conferencing over packet-switched networks that don't provide guaranteed quality of service, such as the Internet. It is an umbrella system which includes the following [3][8]:

- H.225 – call control signaling, registration and admission
- H.235 – security issues
- H.245 – channel usage negotiation
- H.261 – Video Codecs
- G.723 and G.729 – Audio Codecs

6. Session Initiation Protocol (SIP)

SIP, an Internet Engineering Task Force (IETF) standard, is a signaling protocol specifically designed for VoIP. SIP is an application-layer protocol for creating, modifying and terminating sessions with one or more participants [9]. Compared to H.323, SIP is lightweight and simpler to implement.

7. H.235

ITU H.235 is a standard of Security and Encryption for H-Series (H.323 and other H.245-based) Multimedia Terminals [10]. It is intended to provide authentication and encryption service for both point-to-point and multi-point conferences. It is applicable to any terminals utilizing other H. series protocols and H.245 as their control protocol. H.235 has the ability to negotiate services and select various cryptography techniques for specific applications. It supports IPsec, AES, password-based and sophisticated digital

signature, and Elliptic Curve authentication and key agreement. Re-keying may be achieved when synchronization is lost or loss of privacy is detected.

2.1.2 Voice Coding

Voice signals are analog signals and must be converted into digital signals for transmission. Analog/Digital conversion can be accomplished by periodically sampling the analog signals. For example, a waveform can be sampled constantly and then the samples are quantified and represented in a number of binary bits. The number of binary bits generated per second is called bit rate. Because bit rate is very important in voice coding, various coding techniques are developed to achieve low bit rate, but at the cost of delay, complexity and quality of service.

ITU-T G.711 Pulse Code Modulation (PCM) [11] is the simplest speech codec specification with the highest bit rate of 64kb/s and sampling frequency of 8000 Hz. Support of G.711 is usually mandatory. Linear predictive analysis is a powerful technique to estimate voice parameters. Its efficiency can be accomplished by using preceding samples to extract useful information. Typical linear predictive codecs are G.723.1 (5.3kb/s and 6.3kb/s) and G.729A (8kb/s).

2.1.3 Characteristics of VoIP

1. *Speech Packetization*

To accomplish the transmission over IP networks, digital voice data must be packetized in a certain format. For instance, a commonly used PCM sampling frequency is 8000 samples per second, which means a speech of one-second length can be represented by 8000 samples. Then a number of samples are grouped as a frame; coding

algorithms may also apply according to different applications. Then several frames are assembled to form a packet. The size of a packet is determined by the size of frame, sampling frequency, the coding algorithm used and the number of frames in a packet.

Before a voice packet can be transmitted over an IP network, it has to be encapsulated into an IP packet. A typical VoIP application uses UDP and RTP to improve delivery of voice packets. As illustrated in Fig. 2.1, the voice packet itself is the payload; a 12-byte RTP header is added to it. Then an 8-byte UDP header with source and destination socket numbers is added. Finally, a 20-byte IP header including source and destination IP addresses is added.

IP header 20 bytes	UDP header 8 bytes	RTP header 12 bytes	Voice packet payload
-----------------------	-----------------------	------------------------	-------------------------

Figure 2. 1 Format of Voice Packet

There are 40 bytes overhead for a voice packet in total. It is obvious that the larger the size of a voice packet, the higher the efficiency of bandwidth utilization because the header has less portion of a packet. But this doesn't mean the bigger the ratio (payload to overhead) the better the performance. First of all, from the point of view of minimizing packetization delay, the voice packet should be as short as possible – it takes less time to collect fewer samples in packet generation. Secondly, a smaller packet is desirable for better quality of service. For example, there are two voice packets; one is 20 ms and the other is 50 ms. If, for some reason, both of them are lost, then voice gaps of 20 ms and 50 ms take place, respectively. It is not hard to understand that the 20 ms voice gap is less noticeable to the listener and easier to conceal, compared to the 50 ms voice

gap. Thirdly, shorter packet length introduces lower loss rate, as claimed in [12][13]. Therefore, compromise in choosing packet size is necessary in terms of efficiency and performance. Conventionally, a packet size of 20 ms - 40 ms is selected; a packet size over 60 ms usually will not satisfy QoS requirements.

2. Delay

As mentioned earlier, due to the nature of an IP network, UDP packet delivery is not guaranteed and packet delay is not avoidable. One-way end-to-end delay is the amount of time measured from the moment the speaker begins to talk until the listener hears it. Round trip delay is the sum of the two one-way delays and comprises the call [14]. The total delay usually includes the following:

- Encoding and decoding delay
- Compression and decompression delay
- Voice packetization/depacketization delay
- Queuing delay
- Transmission delay
- Propagation delay
- Encryption and decryption delay (if applicable)

Some of the delays are relatively fixed delays, e.g. propagation delay. Transmission delay and propagation delays are so small compared to other delays that people usually tend to ignore them in their investigations.

In general, the less delay, the better QoS. According to ITU-T Rec. G.114 [15], a one-way delay of less than 150 ms is acceptable for most applications and over 400 ms is unacceptable for general networking planning purposes. [16] stated that the most critical

users required a delay of 200 ms or less, more tolerant users were satisfied with a delay of 300 ms – 800 ms. [17] pointed out that studies had shown that a delay of 300 ms may be the upper limit for one-way packet voice in a network.

Table 2. 1 End-to-end VoIP Packet Delays

Latency delay source	Typical delay (ms)
Recording	10 - 40
Encoding (codec)	5 - 10
Compression (speech coder)	5 - 10
Internet delivery	70 - 120
Jitter buffer	50 - 200
Decompression (speech coder)	5 - 10
Decode (codec)	5 - 10
Average	150 - 400

Table 2.1 shows typical end-to-end VoIP packet delays published by Hitachi Semiconductor (America) Inc. and quoted by [18][19].

3. Delay Jitter

The unguaranteed UDP packet delivery leads to varying packet arrival rates, so the voice quality is affected by delay and delay jitter [20]. Although the time intervals of generating and transmitting voice packets are consistent at the source end, the time intervals between arriving voice packets are usually changing.

Given two voice packets, the difference between packet inter-arrival time and packet inter-departure time is called jitter or delay jitter. The inter-arrival jitter between the i^{th} and j^{th} packets denoted as $J(i, j)$ can be expressed as the following equation [12]:

$$J(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (\text{Eq.2.1})$$

where

R_i represents the time of arrival in RTP time stamp for the i^{th} packet, and

S_i represents the time of transmission in RTP time stamp for the j^{th} packet.

Delay jitter should be within a certain range so as to produce a smooth play out stream at the receiver end [21]. Figure 2.2 illustrates the varying time intervals between arriving packets and the play out stream that has the jitter removed. While replaying voice packets, the play out time interval between packets should be kept the same as their inter-departure time, otherwise the played out speech is distorted. So given a certain play out delay D , the i^{th} packet has a scheduled time P_i for playing, as follows:

$$P_i = D + S_i \quad (\text{Eq.2.2})$$

If the packet arrived after its scheduled play out time, there is a packet loss and a replacement packet is played at the scheduled time. In Fig. 2.2, we observe that since packet #5 arrives after its schedule P_5 , it is dropped and replaced by a make-up packet (in shadow). We may conclude that the larger the play out delay, the fewer packet losses, because more packets arrive “in time”. In order to ensure reasonable voice quality, proper play out delay should be chosen as a compromise with stringent packet loss and latency requirements. There are many play out techniques to improve QoS. [22] surveyed various play out schedulers and [23] compared four play out algorithms. [12] and [24] presented the method of dynamically adjusting play out delay during the voice silence intervals.

Figure 2. 2 Packet Inter-arrival Time and Play Out Scheme Illustration

Jitter is a very important parameter in time-stringent applications in that it may cause an out of sequence packet-stream at the receiving end. Even though a re-ordering scheme is able to solve the problem, it is not an ideal solution for VoIP applications because of the introduced delay. In fact, [17] pointed out that the procedures related to recovery of out of order conditions can be omitted at the network and link layers, for the sake of timeliness.

A jitter buffer is a common approach to compensate for the variation of delay and smooth the voice packet-stream replay. It is used to store incoming voice packets and then play them out at a constant rate. From table 2.1, we know that a typical jitter buffer depth is 50 ms to 200 ms.

4. *Packet Loss*

Because of the re-transmission scheme, TCP is not suitable for real-time applications such as VoIP. UDP is usually adopted for interactive robustness. Unlike

TCP, UDP cannot guarantee packet delivery. Voice packets may be dropped when an IP network is in congestion. Sometimes, even though packets are successfully delivered to the destination, they are discarded if they arrive too late – voice packets are useless for playing if they come later than the packets' scheduled play out time. These dropped packets are also considered packet losses. For those packets that never arrive, their latency can be regarded as infinite.

Packet loss rate is defined as the ratio of lost voice packets to the total number of voice packets in a session. The level of packet loss rate that can be tolerated depends on the nature of the encoding algorithm and on the sampling rate of the voice stream [25]. [14] and [16] claimed that a packet loss rate of less than 10% is generally acceptable.

Although packet loss rate is an important parameter, it doesn't sufficiently describe QoS. Given the same packet loss rate, if packet losses are distributed in short and disjointed segments, the voice heard will have some short clips that may not affect the understanding of the conversation or even be noticed by the listener. On the other hand, if the packet losses are in a burst, a long period of silence will occur, which is not tolerable for the listener. So the length of packet loss segment is also an important parameter to consider in VoIP applications.

2.1.4 Voice Activity Detection (VAD)

PSTN provides guaranteed 64KB links and dedicated connection during the whole conversation session. Because usually more than half of the time of a conversation is in silence, it is not necessary to reserve resources for the silent periods.

VAD is designed to identify silence of speech and to suppress silence packets at

the source end. At the destination end, a Comfort Noise Generation (CNG) scheme is applied to generate random noise, to make the silence intervals sound more natural and avoid having the user mistake the silence for a dead phone line. When VAD is used in VoIP applications, the requirement of detection accuracy is less stringent than that of a Voice Recognition system [26]. As a general rule, an effective VAD algorithm should not introduce much delay, should not require much computation power and in the mean time, keep certain QoS. There are quite a few VAD algorithms with different accuracy in voice detection, as presented in [8][26][27]. The VAD algorithm used in our thesis is energy-based and a fixed voice detection threshold is adopted for simplicity. The voice detection accuracy is not essential in our investigation and is out of the scope of the thesis.

Figure 2.3 shows a typical waveform of a speech recording with many silence intervals. In order to achieve bandwidth utilization efficiency, Voice Activity Detection (VAD) technique is usually implemented to allow omitting these silence intervals.

Figure 2. 3 Waveform of A Sample Speech Recording

Energy-based VAD is an algorithm that identifies talkspurt by computing the energy of each packet and comparing it with certain threshold. If the packet energy exceeds the threshold, it is classified as active voice (talkspurt); otherwise it is classified as inactive voice (silence).

Since voice signals are sampled and assembled into frames and packets, the energy of each sample should be computed first. Suppose a packet consists of m samples, $x(i)$ represents the i^{th} sample of a speech and $E(j)$ represents the energy of the j^{th} packet, then we have [26]:

$$E(j) = \frac{1}{m} \sum_{i=(j-1)m+1}^{jm} x^2(i) \quad (\text{Eq.2.3})$$

2.1.5 Error Control and Concealment Schemes

Voice packets are transmitted over unreliable IP networks; packet losses are very common. The research results of [28] indicated that packet loss is highly bursty, with the majority of individual losses occurring in a relatively small number of bursts. This characteristic leads to intolerable clips in the voice stream and degrades the quality of service. In order to deal with this kind of problem, a number of error control and concealment schemes have been developed.

1. *Forward Error Correction (FEC)*

FEC schemes can be used to minimize packet losses by transmitting redundant information of the packets. [29] introduced a media-specific FEC scheme that can recover lost packets by transmitting multiple packets of each voice unit. The disadvantages are the increased delay and bandwidth utilization.

2. *Frame Interleaving*

A frame interleaving scheme is carried out by redistributing adjacent frames to different packets, thereby converting bursty losses to random losses that are much easier to recover [30]. With this scheme, a loss segment can be split into smaller disjointed pieces, which is desirable for the QoS of VoIP. Of course, frame re-ordering is necessary before play out at the receiving end. Frame interleaving introduces overheads when constructing packets and re-ordering frames.

Let k be the number of frames in a packet and m be the space of interleaved frames. Figure 2.4 illustrates the effects of the frame-interleaving scheme on packet losses with $k=4$ and $m=3$. For example, if packet 2 is lost, the four frames that can't be

played out correctly are frame 2, 5, 8 and 11. In the re-ordered stream, the length of loss is reduced from the size of one packet to $\frac{1}{4}$ of a packet.

Figure 2. 4 Frame Interleaving Scheme

3. Error Concealment

Error concealment schemes recover lost packets by generating replacements similar to the original. Because of the redundancy of voice, VoIP can tolerate some distortions without annoying the listener too much within certain levels of packet loss rate ($\leq 15\%$), loss segment length (less than typical length of phoneme) and for small packets (4-40 ms) [29].

There are quite a few error concealment schemes available. Here we simply introduce four insertion-based concealment schemes [29]:

- Splicing

Lost packets are ignored and received packets are spliced without any gaps. Suitable for a low packet loss rate of less than 3% and short loss segment length (4- 16 ms).

- **Silence Substitution**

Lost packets are replaced by silence. It is more suitable for a low packet loss rate of less than 2% and short loss segment length (4 ms). Widely used due to its simplicity of implementation.

- **Noise Substitution**

Lost packets are replaced by background noise. Better quality can be achieved compared to Silence Substitution.

- **Repetition**

Lost packets are replaced by repeating the preceding packet or even by gradually the fading repeated packet. It performs well compared to the three schemes above.

2.2 Cryptography and Cryptographic Systems

2.2.1 Basic Cryptographic Concepts and Requirements

Cryptography is utilized to provide secure information. It is the technique of covering the content of messages by a code or a cipher [31]. Only the intended participants are able to recover the original messages and unauthorized parties are prevented from discovering the secret. Cryptanalysis is the technique of disclosing the secret messages without sufficient knowledge. The original message is called plaintext, and the disguised message is called ciphertext. Encryption or enciphering is the process of converting a plaintext to a ciphertext. The reverse process is decryption or deciphering [32]. The algorithm of cryptography is called cipher.

Analogue protection schemes are not very effective and other complex approaches based on the frequency domain are weak and expensive and can distort the original signal. Therefore, digital cryptographic methods are usually implemented for their simplicity and security. [33]

A cryptographic system (cryptosystem) can be described as:

where f is the encryption algorithm and f^{-1} is the decryption algorithm.

In general, the basic security services required to protect a network are as follows [34][35][36]:

- *Authentication*: verifying the identities of the involved entities (data origin and the peer entity)
- *Access Control*: defining the rules for authorized users to access specific objects
- *Confidentiality*: protecting the data
- *Integrity*: ensuring the received information is the original message (neither altered, destroyed, created, replayed, duplicated nor reordered)
- *Non repudiation*: the ability to prove transmission and receipt of information

Media stream encryption algorithms are chosen according to the following four important criteria as stated in [37]:

- Error propagation and recovery, and their effects on bandwidth consumption
- Performance and cost
- Strength of encryption
- Availability (intellectual property and import/export restrictions)

2.2.2 Cryptosystem Implementation Options

Encryption can be carried link-by-link or end-to-end. The latter implementation is preferable in VoIP, taking security and delay into account.

In the link-by-link encryption, encrypted data should be decrypted by a receiving node and then re-encrypted and sent to the next node. This procedure is repeated on each node of the communication path. It requires extra resources such as memory and CPU for encryption and decryption, and key management, thus introduces network latency. The most serious disadvantage is that the plaintext exists on each node for a certain period of time, so it gives adversaries more opportunities to disclose the secret messages. It also assumes that all nodes involved are trusted entities because they all hold the key and can access the plaintext, which is neither realistic nor easy to manage. Therefore, link-by-link encryption is not ideal in terms of security. From the point of view of VoIP - a time-constrained application - extra latency of encryption/decryption is definitely undesirable.

In the end-to-end encryption, messages are encrypted at the source end and decrypted at the destination end, only the ciphertext is transmitted over the network. Whenever data are traversing over the network, the security level heavily depends on how secure the cryptographic algorithm and key management method are. There is no more trust assumption on the nodes-in-the-middle.

Since most encryption devices encrypt whatever they get, the best way is to encrypt the traffic before sending it to the network. The network can create a control packet around the encrypted traffic [38]. For instance, the voice packet headers don't need encryption and the encrypted voice is the payload.

2.2.3 Stream Cipher and Block Cipher

There are two types of cryptographic algorithms: stream cipher and block cipher. Stream cipher algorithms encrypt a string of data bit by bit, whereas the block cipher algorithms encrypt a certain number of bits each time. Padding is necessary when the last block is shorter than the given length of the block [31].

In case errors occur during transmission, data encrypted with a stream cipher will have errors in the decrypted text exactly when transmission errors occurred. But if a block cipher has a single bit error, the decrypted text is entirely random. Stream ciphers typically operate on small units of plaintext and can be designed to be extremely fast, much faster than any block cipher [39].

[40] explained a public key stream cipher (Quadratic Residue Cipher) suitable for voice packets. However, there is no de facto standard for stream cipher today. Block cipher is commonly used and there are some popular standards and algorithms such as Data Encryption Standards (DES), Advanced Encryption Standard (AES) and de facto standard Rivest-Shamir-Adleman (RSA), etc.

2.2.4 Types of Cryptosystems

Most encryption algorithms involve one or more keys, so they are usually classified according to the type of keys being used. There are two types of cryptosystems: secret (symmetric) key cryptosystems and public (asymmetric) key cryptosystems.

2.2.4.1 Secret Key Cryptosystems

A secret key cryptosystem uses a shared key among participants. The key is used in the encryption algorithm to convert plaintext into ciphertext and in the decryption

algorithm to recover plaintext from ciphertext. The key size is chosen such that an exhaustive key search is considered infeasible, taking into account the newest analysis and technology. One of the most important issues is how to distribute the secret key securely. If keys are easily compromised during distribution, then the security is very weak, no matter how strong the encryption algorithm is.

1. Data Encryption Standard (DES)

The most well known and widely implemented secret key algorithm is the Data Encryption Standard (FIPS PUB 46), published by NIST in 1977, and re-certified a couple of times. Figure 2.5 describes the algorithm of DES.

DES has shown strong security in the past two decades. As specified in [41], DES is a 64-bit block cipher encryption algorithm. A key consists of 64 binary digits of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits are used for error detection (so the length of the key is 56 bits and the remaining 8 bits are set to make the parity of each 8-bit byte of the key odd).

The DES algorithm is designed to encrypt and decrypt 64-bit blocks of data under the control of a 64-bit key. A block to be encrypted is subjected to an initial permutation, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation. Shown in Fig. 2.5, DES defines 16 iterations for the encryption computation and applies on one half of the data. And in each iteration, a different 48-bit block sub-key is derived from the 64-bit key being used.

Figure 2. 5 DES Encryption Computation

Let E and D represent encryption and decryption algorithms, respectively. K is the secret key, P is plaintext and C is ciphertext. Let $L_i R_i$ be the 64-bit permuted input block, L_i and R_i are 32-bit blocks and K_{i+1} is a 48-bit sub-key derived from the session key. The output $L_{i+1} R_{i+1}$ is defined as:

$$L_{i+1} = R_i \quad (\text{Eq.2.4})$$

$$R_{i+1} = L_i \oplus f(R_i, K_{i+1}) \quad (\text{Eq.2.5})$$

To decrypt the blocks, using the inverse permutation and the very same algorithm that is applied, each iteration of the computation uses the correspondent sub-key.

$$R_i = L_{i+1} \quad (\text{Eq.2.6})$$

$$L_i = R_{i+1} \oplus f(L_{i+1}, K_{i+1}) \quad (\text{Eq.2.7})$$

According to [31], one should note that there are four weak and twelve semi-weak keys while operating the DES algorithm. Weak keys occur because of the way bits are selected from the encryption key to construct the sub-key ($E_K(E_K(P)) = P$) and semi-weak keys occur because of the key's analogous property ($E_{K1}(E_{K2}(P)) = P$). Because keys are generated at random, the probability of getting a weak key is very low. However, these known weak and semi-weak keys may be avoided [39]. There are over 7×10^{16} possible DES keys of 56 bits. In order to prevent potential attacks, users should change key frequently. "When used properly, DES is secure against all but the most powerful enemies. In fact, triple encryption DES may be secure against anyone at all." [39].

2. Triple Data Encryption Algorithm (TDEA)

[41] defines Triple Data Encryption Algorithm (TDEA). Each TDEA encryption/decryption operation is a compound of DES operations. It applies DES three

times and the key size is $56 \times 3 = 168$ bits, which provides enforced security.

TDEA encryption/decryption operations are:

$$C = E_{K3} (D_{K2} (E_{K1} (P))) \quad (\text{Eq.2.8})$$

$$P = D_{K1} (E_{K2} (D_{K3} (C))) \quad (\text{Eq.2.9})$$

Figure 2. 6 TDEA Encryption/Decryption Operations

Figure 2.6 depicts the TDEA cryptographic system. There are three options for key selection:

- $K1, K2, K3$ are independent
- $K1, K2$ are independent and $K1 = K3$
- $K1 = K2 = K3$ (as single DES)

TDEA has seven approved modes of operation specified in ANSI X9.52. Four of them are equivalent to ECB, CBC, CFB and OFB modes in DES. The remaining three are mode variants that allow for interleaving or pipeline [42].

2. Advanced Encryption Standard (AES)

With the development of cryptanalysis techniques and the availability of powerful computers, 64-bit DES is not as secure as before. The Advanced Encryption Standard (AES) is a new secret key encryption standard specified by NIST, which was put into

effect in May 2002. [41] stated that AES is intended to replace DES and TDEA to protect electronic data, but it anticipated TDEA and AES would coexist for gradual transition.

AES specifies the Rijndael algorithm, which is a block cipher using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. According to [43], at the start of the encryption procedure, the input data are copied to the State array according to a scheme. After an initial Round Key addition, the State array is transformed by implementing a round function 10, 12, or 14 times, depending on the key length, with the final round differing slightly from the previous rounds. The sub-keys are generated from the input key by a Key Expansion routine. The decryption procedure is a straightforward inverse of the encryption algorithm. Unlike the DES algorithm, no weak or semi-weak keys have been identified for the AES algorithm, and there is no restriction on key selection.

2.2.4.2 Public Key Cryptosystems

A public key cryptosystem uses a pair of keys, public key and private key, to encrypt and decrypt data. Both of the keys can be used for encryption and decryption. The public key is published and can be obtained by anyone; the private key must be kept secret by the user. It is assumed that the complex of inverse public key to private key prevents the disclosure of the private from the knowledge of the public key.

The Diffie-Hellman algorithm, RSA and Elliptic Curve are well known public key cryptosystems. The margin of security must be taken into account in key length selection [39]. For instance, a 1024-bit long key for RSA is good enough today, but may not be secure a few years from now. In general, key size should be increased gradually so as to protect against continuously developed cryptanalysis techniques.

1. Diffie-Hellman Algorithm (DH)

The Diffie-Hellman algorithm is a method for two parties to exchange secret information without communicating previously. It was published in 1976 and triggered the development of public key algorithms. The DH algorithm can be used for encryption/decryption and authentication. Table 2.2 describes the process of the DH algorithm and there are certain requirements for the numbers used in this algorithm [44].

Table 2. 2 Diffie-Hellman Algorithm [44]

Alice and Bob agree on two numbers “ p ” and “ g ”	“ p ” is a large prime number “ g ” is called the base or generator
Alice picks a secret number “ a ”	Alice’s secret number = a
Bob picks a secret number “ b ”	Bob’s secret number = b
Alice computes her public number $x = g^a \bmod p$	Alice’s public number = x
Bob computes his public number $y = g^b \bmod p$	Bob’s public number = y
Alice and Bob exchange their public numbers	Alice knows p, g, a, x, y Bob knows p, g, b, x, y
Alice computes $k_a = y^a \bmod p$	$k_a = (g^b \bmod p)^a \bmod p$ $= (g^b)^a \bmod p$ $= g^{ba} \bmod p$
Bob computes $k_b = x^b \bmod p$	$k_b = (g^a \bmod p)^b \bmod p$ $= (g^a)^b \bmod p$ $= g^{ab} \bmod p$
By the laws of algebra, $k_a = k_b = k$	Alice and Bob both know the secret value “ k ”

2. RSA

Although there is no primary standard public key cryptosystem, RSA, invented by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977, is considered as a de facto algorithm. In the RSA algorithm, two very large primes p and q are chosen such that

$$n = pq \quad (\text{Eq.2.10})$$

$$m = (p-1)(q-1) \quad (\text{Eq.2.11})$$

where

e is chosen that is less than n and relatively prime to m .

d , the inverse of e , is calculated with mod m .

So the public key is the pair (n, e) , the private key is d . The entire security of RSA depends on the assumption that factoring n to find p and q is very difficult [39].

RSA can be used both for encryption and authentication. It is not necessary for two parties to share any secret before exchanging secure information. If A wants to send a secret message to B , he uses B 's public key to encrypt the message. B decrypts the ciphertext by using B 's private key; nobody else can decrypt it but B because he is the only one holding the private key. If A sends a message encrypted by using A 's private key, the message can be authenticated by the verifier using A 's public key to decrypt it.

2.2.5 DES Modes of Operation

Because the DES algorithm is so widely implemented, it is very important to understand the characteristics of its operation. [42] defines five modes of operation for the DES algorithm. The modes included in this standard are the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CFB)

mode, the Output Feedback (OFB) mode and the Counter (CTR) mode. As mentioned earlier, padding is necessary for all the operation modes.

1. Electronic Codebook (ECB) mode

The Electronic Codebook mode is defined as a confidentiality mode, where each plaintext block is encrypted independently and the ciphertext block is decrypted independently. For a given key, a fixed plaintext block corresponds to a fixed ciphertext block, analogous to the assignment of code words in a codebook. ECB mode is illustrated in Fig. 2.7 and is defined as follows:

$$\text{Encryption:} \quad C_i = E_K(P_i) \quad (\text{Eq.2.12})$$

$$\text{Decryption:} \quad P_i = D_K(C_i) \quad (\text{Eq.2.13})$$

where i represents the i^{th} block and $i=1, 2, 3, \dots, n$.

Figure 2. 7 The Electronic Codebook Mode

On the one hand, as claimed in [31], the weakness of the ECB mode lies in that it doesn't connect the blocks together, therefore an adversary can reconstruct messages from the parts of the messages he had intercepted. Additionally, a given plaintext always produces the same ciphertext with the same key, which puts the secret in risk of disclosure. On the other hand, an error in transmitting a ciphertext block doesn't affect

the decryption of successive blocks at all. This independent characteristic of ECB mode is definitely desirable.

2. Cipher Block Chaining (CBC) mode

The Cipher Block Chaining mode is defined as a confidentiality mode, where the encryption of a plaintext block makes use of its previous ciphertext block, and so does the decryption of a ciphertext block. The chaining ensures that identical plaintext blocks don't produce the same ciphertext, for stronger security. CBC mode requires a random Initialization Vector (IV) for the first block.

The CBC mode is described in Fig. 2.8 and can be expressed as the following:

$$\text{Encryption:} \quad C_1 = E_K(P_1 \oplus IV) \quad (\text{Eq.2.14})$$

$$C_i = E_K(P_i \oplus C_{i-1}) \quad (\text{Eq.2.15})$$

$$\text{Decryption:} \quad P_1 = D_K(C_1) \oplus IV \quad (\text{Eq.2.16})$$

$$P_i = D_K(C_i) \oplus C_{i-1} \quad (\text{Eq.2.17})$$

where i represents the i^{th} block and $i = 2, 3, \dots, n$.

In CBC mode, provided that all ciphertext is correctly computed, if an encrypted block is lost or in error, the next block couldn't be decrypted correctly. So the encryption and decryption of a block depend on the previous block ciphertext. CBC mode has self-recovering ability [31]; whenever an error occurs in an encrypted block, those blocks after the succeeding block are not affected. As shown in Fig. 2.9, let n be the number of blocks in a packet and $b_{i,j}$ represent the j^{th} block of the i^{th} packet. Then the last block of the i^{th} packet ($b_{i,n}$) is the decryption input for the first block of the $(i+1)^{\text{th}}$ packet ($b_{i+1,1}$). If packet 2 is dropped (in shadow) due to delay, then $b_{3,1}$ can't be decrypted correctly because of the lack of block $b_{2,n}$, but the subsequent blocks after $b_{3,1}$ are not affected.

Therefore, when CBC mode is used in VoIP, error propagation doesn't extend beyond the next voice packet.

Figure 2. 8 The Cipher Block Chaining Mode

Figure 2. 9 Error Propagation of the CBC Mode

Some literature suggested that the whole chain should be discarded and retransmitted to achieve high security whenever errors are detected in a chain. But for VoIP applications, time-consuming retransmission may not be acceptable, so only blocks in error should be discarded or maybe simply ignored until the procedure recovers itself. However, a threshold for error may be set up to monitor if the call is at risk of attack.

3. Cipher Feedback (CFB) mode

The Cipher Feedback mode is defined as a confidentiality mode, where the encryption of an s -bit plaintext block makes use of the previous s -bit ciphertext block as part of the b -bit encryption input ($b \geq s$), and so does the decryption of a ciphertext block. A b -bit shift register that stores the encryption input is used to shift in s -bit new feedback ciphertext and shift out s -bit old feedback ciphertext each time. CFB mode encrypts or decrypts s -bit plaintext at a time, so its throughput is less than ECB mode and CBC mode when $s < 64$ [17].

The CFB mode is defined as the following:

$$\text{Encryption:} \quad I_1 = IV \quad (\text{Eq.2.18})$$

$$I_i = \text{LSB}_{b-s}(I_{i-1}) \mid C_{i-1} \quad (\text{Eq.2.19})$$

$$O_i = E_K(I_i) \quad (\text{Eq.2.20})$$

$$C_i = P_i \oplus \text{MSB}_s(O_i) \quad (\text{Eq.2.21})$$

$$\text{Decryption:} \quad I_1 = IV$$

$$I_i = \text{LSB}_{b-s}(I_{i-1}) \mid C_{i-1}$$

$$O_i = E_K(I_i)$$

$$P_i = C_i \oplus \text{MSB}_s(O_i) \quad (\text{Eq.2.22})$$

where i represents the i^{th} block and $i = 2, 3, \dots, n$.

Similar to CBC mode, CFB mode needs an IV and the encryption and decryption of a block depend on the segment of previous ciphertext block. The CFB mode is described in Fig. 2.10. The only difference is that if an encrypted block is lost or in error, the next a few blocks can't be decrypted correctly. The size of the feedback ciphertext s (i.e. the size of plaintext block) and the size of the encryption input b are optional.

Figure 2. 10 The Cipher Feedback Mode

CFB mode is also a self-recovering mode [31]. Figure 2.11 illustrates its error propagation characteristic. Let m be the number of feedback blocks in the shift register (i.e. encryption input); b , i , and n are defined the same as for CBC mode, and we have:

$$b = ms \quad (\text{Eq.2.23})$$

So the last block of the i^{th} packet ($b_{i,n}$) is the feedback kept in the shift register to decrypt the blocks from $b_{i+1,1}$ to $b_{i+1,m}$ of the $(i+1)^{th}$ packet. If packet 2 is dropped (in shadow) due to delay, then block $b_{3,1}, b_{3,2}, \dots, b_{3,m}$ can't be decrypted correctly because of the lack of $b_{2,n}$, but the subsequent blocks after $b_{3,m}$ are not affected. n is not necessarily larger than m , thus an error may propagate more than one packet. But in practice, it is proper to limit the error propagation within the next packet so as to reduce packet loss.

Figure 2. 11 Error Propagation of the CFB Mode

4. Output Feedback (OFB) mode

The Output Feedback mode is defined as a confidentiality mode that uses previous middle-ciphertext as feedback to modify the input block. An IV is required for the first block and can be used only once for a given key.

The OFB mode is defined as the following:

Encryption:

$$I_1 = IV$$

$$I_i = O_{i-1} \quad (\text{Eq.2.24})$$

$$O_i = E_K(I_i) \quad (\text{Eq.2.25})$$

$$C_i = P_i \oplus O_i \quad (\text{Eq.2.26})$$

Decryption:

$$I_1 = IV$$

$$I_i = O_{i-1}$$

$$O_i = E_K(I_i)$$

$$P_i = C_i \oplus O_i \quad (\text{Eq.2.27})$$

where i represents the i^{th} block and $i = 2, 3, \dots, n$.

Figure 2. 12 The Output Feedback Mode

From Fig. 2.12, we observe that the feedback is independent of the plaintext. OFB mode avoids the effect of the final results of the previous ciphertext block, therefore an error in one block wouldn't extend to other blocks. However, a good synchronization scheme is essential when a ciphertext block deletion or insertion occurs; otherwise, the result of the decryption is random [31]. Nevertheless, OFB mode hasn't the problem as in

CBC and CFB modes when error occurs, which may benefit the encryption of VoIP. Literature [45] stated that OFB mode is implemented in a commercial security system because of its immunity to bit errors likely to occur in the noisy communication channel.

5. Counter (CTR) mode

The Counter mode is defined as a confidentiality mode that uses counter content to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. For a given key, the sequence of the counter for each block must be distinct to protect against 'codebook' attack. The CTR mode is illustrated in Fig. 2.13 and is defined as the following:

$$\text{Encryption:} \quad O_i = E_K(T_i) \quad (\text{Eq.2.28})$$

$$C_i = P_i \oplus O_i \quad (\text{Eq.2.29})$$

$$\text{Decryption:} \quad O_i = E_K(T_i)$$

$$P_i = C_i \oplus O_i \quad (\text{Eq.2.30})$$

where T_i is the counter value and i represents the i^{th} block and $i = 1, 2, 3, \dots, n$.

Similar to OFB mode, with CTR mode employed, a plaintext block can be encrypted and a ciphertext block can be decrypted without other ciphertext blocks. CTR mode also suffers from a synchronization problem if the counter is out of sequence.

Figure 2. 13 The Counter Mode

In summary, CBC and CFB modes both involve ciphertext block feedback, therefore errors might propagate and the cryptosystem would encounter more decrypting errors and the original voice packets couldn't be reconstructed properly. ECB mode is rather weak in terms of security; OFB and CTR modes are not ideal for voice because of the difficulties in synchronization. Proper encryption mode should be selected so as to meet the stringent constraints of VoIP in terms of packet delay and packet loss. For these reasons, CBC mode is mostly implemented and sometimes it is even mandatory. Although CFB mode is a stronger mode, it suffers from more error propagation and is not used as often as CBC mode. In fact, CBC mode can be treated as a special case of CFB

mode. Therefore, in practice, the most important issue of secure real-time service is how to deal with error propagation so as to achieve low packet losses.

2.2.6 Comparisons of Secret Key Cryptosystem and Public Key Cryptosystem

Both Secret Key Cryptosystems and Public Key Cryptosystems have pros and cons. On the one hand, compared to a secret key cryptosystem, the length of key of a public key cryptosystem is much longer - usually over a thousand bits. Also its encryption/decryption speed is much slower. According to [39], DES is at least 100 times as fast as RSA in software, and 1000 to 10000 times as fast as RSA in hardware. [46] presented an implementation of the DES algorithm for real-time application and [17] claimed that the DES algorithm usually introduced a few microseconds delay in encryption/decryption, where the public key algorithm may introduce one-third of a second delay, which is unacceptable in real-time applications. Moreover, because of the size of the key, more resources are required for the encryption/decryption and key management in public key cryptosystems.

On the other hand, the public key cryptosystem is more suitable in the circumstance that either secure or face-to-face shared secret exchange is difficult and inconvenient. The only necessary information is the published public key. Because the private key is never shared or transmitted over the network, key compromising is less possible and a higher level of security can be achieved. Although the public key cryptosystem is slower than secret key cryptosystem, it can provide reliable and convenient secret key distribution. By using the public key cryptosystem for key management and authentication, and using the secret key cryptosystem for data encryption and decryption, a combination of the two cryptosystems can make use of both

benefits.

2.2.7 Issues of Secure VoIP

H.323 and SIP are popular protocols for VoIP communications. Various features of these protocols require different secure mechanisms. However, the characteristics of VoIP indicate some common issues to be considered in secure VoIP applications. For instance, how to protect a call against man-in-the-middle, how to achieve authentication in arbitrarily large networks [46], how to provide secure connections with limited resources, and how to minimize the impact on QoS as typical cryptosystems are selected. It is well known that latency is of great importance for VoIP. The real-time nature of voice packets determines that the cryptosystems used in VoIP applications have to sacrifice some reliability so as to meet the time constraint. Error recovery and flow-control mechanisms must be limited as well. Keeping these issues in mind may give us some general guidelines in implementing secure VoIP applications.

Chapter 3 The Multiple Packet-Stream Scheme

DES is the widely employed cryptosystem in media cryptography and its modes of operation with feedback are commonly implemented for security purposes. Although a longer key is usually a solution against cryptanalysis, it requires more memories and computation power. Moreover, because packet latency and packet losses are very common in network communications, and error control and retransmission are limited in interactive applications, the lack of reliable transmission may further degrade the QoS of VoIP due to the error propagation characteristics of CBC and CFB modes.

In this thesis, we propose a multiple packet-stream scheme that breaks a voice stream into several streams, allows using multiple secret keys in voice cryptography and can reduce error propagation when packet latency and loss occur. To the best of our knowledge, we haven't found any literature addressing using multiple packet-streams in encrypted VoIP with CBC mode or CFB mode implementation.

3.1 Encrypted Packet-Stream Scheme

A single packet-stream can be re-organized so as to construct several streams, for example, three streams. In the following discussion, we assume that there are not computation errors or transmission errors, packet losses are due to latency and an error only propagates to the next packet as with the CBC confidentiality mode.

A packet in the multiple packet-streams can be defined as

$$P_n(i, j) = P_o[(j - 1)n + i], \quad 1 \leq i \leq n, \quad (\text{Eq.3.1})$$

where n represents the number of packet-streams being used, $P_o(m)$ denotes the m^{th}

packet of the original single packet-stream, and $P_n(i, j)$ denotes the j^{th} packet in the i^{th} packet-stream. While encrypting voice packets, an IV is used by the first plaintext packet of each stream and each ciphertext packet is used as the encryption input for its subsequent plaintext packet. While decrypting voice packets, an IV is used by the first ciphertext packet of each stream and each ciphertext packet is used as the decryption input for its subsequent ciphertext packet.

Figure 3. 1 The Structure of Multiple Packet-Stream Scheme

Since the encryption process doesn't encounter any error with our assumptions, we only focus on discussing the decryption process. Figure 3.1(a) is the original packet-

stream encrypted with encryption key k . In Fig. 3.1(b), $n=3$, the first stream contains packet #1, #4, #7, #10, ... ; the second stream contains packet #2, #5, #8, #11, ... ; the third stream contains packet #3, #6, #9, #12, And k_1 , k_2 and k_3 are encryption keys for Stream #1, #2 and #3, respectively.

Let T_p be the constant packet-generating time interval between two adjacent packets in the original single packet-stream. Then the play out time interval between the two packets must be T_p , so as to replay the message at the same speed as was recorded. For any two packets r and m in the original stream, their packet generating time interval or play out time interval is:

$$T[P_o(m)] - T[P_o(r)] = (m - r) T_p \quad 1 \leq r \leq m, \quad (\text{Eq. 3.2})$$

where $T[P_o(m)]$ is the packet generating time of the m^{th} packet.

Similarly, let $T[P_n(i, j)]$ denote the packet-generating time of the j^{th} packet in the i^{th} packet-stream of n multiple packet-streams; the packet-generating time interval T_{pn} between two adjacent packets can be defined as:

$$T_{pn} = T[P_n(i, j+1)] - T[P_n(i, j)] \quad (\text{Eq. 3.3})$$

According to Eq.3.1, Eq.3.2 and Eq.3.3, we can derive that

$$\begin{aligned} T_{pn} &= T[P_o(jn + i)] - T[P_o((j-1)n + i)] \\ &= nT_p \end{aligned} \quad (\text{Eq. 3.4})$$

From Eq.3.4, it is clear that while using n multiple packet-streams, the play out time interval between two adjacent packets of each packet-stream increases linearly by n . So the acceptable packet for decryption must arrive no later than nT_p .

3.2 Benefits of Multiple Packet-Stream Scheme

3.2.1 Reduced Error Propagation Effects

The multiple packet-stream approach minimizes the cryptographic impact of error propagation by providing necessary decryption information for the next coming packet and in consequence improves QoS. The more packet-streams being used, the fewer packet losses due to error propagation.

1. *Relaxed Time Constraints*

In Fig. 3.1(a) the original single packet-stream is shown. According to Eq.3.4, each packet is the decryption input of its subsequent packet with the acceptable arrival delay up to T_p . For example, packet #7 is the decryption input of packet #8. In Fig. 3.1(b) with $n=3$, we observe that packet #7 in Stream #1 is the decryption input for packet #10 with acceptable arrival delay up to $3T_p$. This delay limit applies to all other packets.

Notice that in this scheme, T_{pn} can relax the time constraint for the decryption process since it allows more time for a late packet to arrive. If a packet is too late for play out (i.e., arrives after T_p) but is still early enough for decryption (i.e., arrives before T_{pn}) then the packet is kept so that the next packet in that stream can still be decrypted without loss. In this way, the decryption failure for CBC and CFB modes can often be avoided.

2. *Shortened Length of Packet Loss Segment*

Short packet loss segments are more tolerable and can provide better voice quality than long packet loss segments. In the single stream shown in Fig. 3.2, let packet #1 and #2, packet #4 and #5, packet #7 and #8 be three loss segments due to latency. Then Packet #3 is a loss due to the error propagated from packet #2. Similarly, packet #6 is a loss that propagated from packet #5 and packet #9 is a loss that propagated from packet

#8. So there are totally 9 packet losses in one loss segment. When three packet-streams are chosen, in Stream #1, the lost packet #1 propagates an error to packet #4, the lost packet #4 propagates an error to packet #7 and the lost packet #7 propagates an error to packet #10; in Stream #2, the lost packet #2 propagates an error to packet #5, the lost packet #5 propagates an error to packet #8 and the lost packet #8 propagates an error to packet #11. There are totally 8 lost packets distributed in four segments with length of 2 packets each. Therefore, a loss segment can be broken into smaller pieces that help the listener to better understand the recovered message.

Figure 3. 2 Effect of Multiple Packet-Stream Scheme on Packet Loss

We also note that the extra resource the scheme requires is a larger buffer that depends on the number of packet-streams, because those packets that have been played out should be kept in the buffer until they are used as decryption inputs.

3.2.2 Enhanced Message Security

Whenever the multiple packet-stream scheme is adopted, the packets are generated the same way as in the single-stream without the proposed scheme. Instead of using a single secret key for encryption at a time, each packet-stream may have its own secret key and a certain security level can be achieved. Of course frequent key changing is necessary for each stream to better secure the conversation. To decrypt a packet-stream, the same secret key must be used and it is rather simple to determine the decryption key by finding out which stream a packet belongs to. A packet-stream ID may be added to each packet so as to indicate the stream number or just simply the modulus may be taken of the original packet sequence number by the number of packet-streams being used.

There are a few advantages of our proposed scheme in terms of maintaining message privacy. Firstly, while using different secret keys for each packet-stream, in case a secret key is compromised, only the stream encrypted by the compromised key is disclosed, but not the whole message. Secondly, because all the packets within one stream are not really adjacent – they are at a distance from each other (i.e. the number of packet-streams), the disclosed part of a message only consists short and disjointed information pieces. This makes it relatively more difficult for an eavesdropper to understand the real content. In contrast, the frequent key change method for single-stream will leak a chunk of original message if a key is broken.

Referring to Fig. 3.1, the original packet-stream includes m packets and is encrypted by a secret key k . It can be split into n packet-streams so that each of the streams includes m/n packets and the streams are encrypted by $k_1, k_2, \dots$ and k_n ,

respectively. These n secret keys are equivalent to k in terms of strength of security. If one key is broken, the comparisons for with or without the multiple packet-stream scheme are given in table 3.1.

Table 3. 1 Effect of Multiple Packet-Streams on Protecting Message Privacy

	Original packet-stream	n packet-streams
Total disclosed packets	m	m/n
Number of disclosed segments	1	m/n
Number of packets in a segment	m	1

From table 3.1, we clearly see that all the m voice packets are disclosed as a chunk piece if k is broken. But if k_i is broken while using n secret keys, only $1/n$ of the voice packets are disclosed, and each disclosure segment has only one packet. Should the attacker want to compromise the whole message, he has to figure out all the secret keys $k_1, k_2, \dots$ and k_n , which requires n times the effort as he did to k . In addition, when n is larger, the disclosed packets are further apart and thereby harder to understand. Finally, the proposed scheme is easy to implement and very flexible in how many secret keys are needed, according to the security requirements. In addition, each packet-stream can be processed on separate computers so as to relax the computation speed requirement.

In the mean time, we point out that some overheads are introduced in multiple-key utilization. It takes extra time to generate and distribute a set of secret keys, and more storage is needed as well. However, compared to the benefits it has, these overheads are tolerable because of stronger security.

Chapter 4 Simulation Setup

Our investigation in this thesis focuses on the encrypted voice packet-stream when the CFB or CBC modes of DES are implemented. We use Matlab to implement the proposed scheme. In our simulation considerations, the following assumptions are made:

- Encryption keys are securely distributed and strong enough against an exhaustive key search
- Successful authentication
- Packets are considered lost due to latency
- No packet transmission errors
- No encryption/decryption processing errors
- Packet headers in plaintext and payload in ciphertext

The simulation structure and other related issues and parameters are described in the following subsections.

4.1 The Simulation Structure

Without the multiple packet-stream scheme, a late packet can't be used for decryption if it is delayed more than T_p compared to its scheduled play out time. To implement this scheme, we use a jitter buffer and a decryption buffer to store arriving packets as shown in Fig. 4.1. Network delays are randomly generated, the packet departure time, arrival time and play out time are calculated; and for a given number of packet-streams, the maximum acceptable delay of each packet for decryption purpose is calculated. Then the packet arrival time is compared to its play out time and maximum

acceptable delay so as to determine if it can be replayed or is just for decryption.

Figure 4. 1 The Buffer Queuing Structure

Normally, the jitter buffer would discard any packet arriving later than its scheduled play out time. In our proposed multiple packet-stream scheme, we modify this behavior as follows:

- If packet arrives before its scheduled play out time and the jitter buffer is not full, place in jitter buffer for play out
- If packet arrives before its scheduled play out time and the jitter buffer is full, place in decryption buffer for play out

- If packet arrives after its scheduled play out time but before T_{pn} (i.e. nT_p), then place packet into decryption buffer
- If packet arrives after T_{pn} , then discard packet

In general, a deeper jitter buffer can better smooth delay jitter and reduce packet losses but at the cost of longer queuing delay. Our approach can reduce packet losses without producing longer queuing delays within the jitter buffer.

Figure 4.2 illustrates the algorithm of the buffers for receiving voice packets. The decryption buffer is used to collect those packets dropped by the jitter buffer because of a certain delay or jitter buffer overflow.

Figure 4. 2 Flowchart of Buffer Receiving/Dropping Algorithm

It should be pointed out that larger buffers are needed for a larger number of packet-streams n because those packets (i.e. n adjacent voice packets) that were played out should be kept in the buffer until they have been used for decryption. In Fig. 4.3, for those stored and played packets in the jitter buffer or decryption buffer, we clearly see that:

- Packets 1 to i are stored in the buffer, when $i \leq n$
- Packets $i-n$ to i are stored in the buffer, when $i > n$

Figure 4. 3 Packet Storage in Buffers with n Multiple Packet-Streams

4.2 Voice Packet Considerations

Since the DES cryptosystem is just a system to disguise digitized voice signals,

the speech coding algorithm should not affect the encryption and decryption. We choose PCM coding for real speech recording. Many studies have shown that a proper voice packet size is usually less than 60 ms. In our investigation, we use 30 ms voice packet and assume that error propagation doesn't exceed one packet. So a delayed packet only affects the next packet.

In our simulations, both real speech samples and three types of exponentially distributed artificial speeches are used, because we feel that people's speaking habits are very different.

1. Real Speech Samples

For real speech recording, energy-based VAD with fixed energy threshold is used. [20] presented two methods to set the threshold. One method trains the VAD algorithm by using a prerecorded sample speech that contains background noise only. The average packet energy is computed as the threshold. The other method is to assume that the initial 200 ms of the sample doesn't contain any speech, but noise. So the average packet energy in this 200 ms interval is calculated as the threshold. We chose method 2 in our simulations.

Let $En(j)$ be the energy of the j^{th} packet and En_thres represent the energy threshold, k is a factor of threshold, for $k > 1$,

if

$$En(j) \geq kEn_thres \quad \text{active voice} \quad (Eq.4.1)$$

else if

$$En(j) < kEn_thres \quad \text{inactive voice} \quad (Eq.4.2)$$

Although the value of k affects the sensitivity of the VAD algorithm, it is out of the scope of this thesis. Figure 4.4 shows the energy-based VAD results of real speech recording.

Figure 4. 4 Energy-based VAD Results ($k=100$)

2. Artificial Speech

Voice consists of talkspurt and silence; this characteristic is often referred to as On-Off patterns. As presented in chapter 2, the On-Off pattern of a real speech can be detected by the VAD algorithm. Traditionally, talkspurt and silence distributions are modeled by exponential distributions. [48][49][50] all quoted Paul Brady's well-known study that led to the popularity of the exponential distribution model. We will adopt this model in our artificial speech generation.

According to [48] and [49], a mean talkspurt of 352 ms and a mean silence of 650 ms are commonly used. The parameters recommended by ITU P.59 for generating artificial human speech are shown in table 4.1.

Table 4. 1 Typical Distribution Parameters of Artificial Speech

	Mean talkspurt	Mean silence
Sriram without hangover ⁽¹⁾	0.352 sec	0.65 sec
Brady with hangover	1.2 sec	1.8 sec
ITU P.59 without hangover	0.227 sec	0.596 sec
ITU P.59 with hangover	1.004 sec	1.587 sec

(1) Hangover is a technique to avoid sudden end-clipping of speeches and to bridge short speech gaps such as those due to stop consonants. [48]

In our artificial speech modeling, we set the generated talkspurts to “ON” or “1” and the silences to “OFF” or “0”, then the results are considered equivalent to VAD output. Figure 4.5 shows two talkspurt/silence simulation results using different distribution parameters. (Type-1: 352 ms / 650 ms; Type-2 : 227ms / 596 ms)

Figure 4. 5 The On-Off Patterns of Artificial Speeches

4.3 Network Delay

The network delay characteristic that affects VoIP applications most is unpredictable. It is widely known that network delay can be classified as long-term and short-term delay.

4.3.1 Long-term Delay

[51] studied the low frequency components of Internet delay and investigated the dominant network frequencies over a working day. It showed that network delay had a strong diurnal pattern. [16] measured the round-trip delay over several Internet paths for months and indicated that network delay follows a diurnal cycle as well.

Figure 4. 6 Typical Mean Delays in a Weekday [16]

Based on common sense, the majority of calls last a few hours, at most. It is very unusual for people to make a phone call lasting longer than a day. The mean delays over a typical path between two universities in a 24-hour weekday are measured and presented in [30]. In our simulations, we used these observed mean delay data as the network long-term delay parameters. They are plotted in Fig. 4.6.

4.3.2 Short-term Delay

1. Long-Range Dependence

Many recent researches on the modeling of network arrivals demonstrated that the network (LAN and WAN) traffic is quite different from the traditional telephone traffic. The correlation of the traffic can extend to large time scales, thus the network traffic has Long-Range Dependence (LDP). In [52], Paxson and Floyd pointed out that the

traditional Poisson process model is not suitable for modern network traffic. They indicated that network traffic is much better modeled as a self-similar process, which is the simplest model displaying LRD characteristics. [53] presented evidence that LRD exists in packet round-trip delays and [54][55] addressed the self-similar nature of LAN and WAN network traffic. Investigation in [56] indicated that self-similar traffic can be constructed by using ON/OFF sources and that the ON and OFF periods are heavy-tailed. A heavy-tailed distribution is defined as:

$$P[X > x] \sim x^{-\alpha} \quad \text{as } x \rightarrow \infty, \quad (0 < \alpha < 2) \quad (\text{Eq.4.3})$$

where the probability is not negligible even as x approaches infinity.

2. Pareto Distribution

Four packet transmission delay models were presented in [57] and Pareto Distribution is the simplest heavy-tailed distribution that is a prevailing model for network traffic. Its probability distribution function is:

$$f(x) = ba^b x^{-b-1} \quad a, b \geq 0, x \geq a \quad (\text{Eq.4.4})$$

Its cumulative distribution function is:

$$F(x) = P[X \leq x] = 1 - (a/x)^b \quad b \geq 0, x \geq a \quad (\text{Eq.4.5})$$

where a , defined as the location parameter, represents the smallest possible random variable; in our case it is the smallest network delay and b is defined as the shape parameter.

The mean and variance of Pareto Distribution are:

$$E(x) = \begin{cases} ab/(b-1) & b > 1 \\ \infty & b \leq 1 \end{cases} \quad (\text{Eq.4.6})$$

$$Var(x) = \begin{cases} a^2 b/(b-2)(b-1)^2 & b > 2 \\ \infty & b \leq 2 \end{cases} \quad (\text{Eq.4.7})$$

$$\infty \quad b \leq 2$$

We chose the shape parameter $1 < b < 2$ so that it has finite mean and infinite variance. Let mean delay $E(x) = \mu$, the location parameter can be derived from Eq.4.6:

$$a = \mu - \mu/b \quad b > 1 \quad (\text{Eq.4.8})$$

From Fig. 4.7 below, we can see that some packets encounter extremely long delay, which is an evidence of the heavy-tailed property.

Figure 4. 7 Pareto Distribution Delays

Because packet loss is often bursty and Pareto Distribution is not ideal to model this characteristic, we also used a sine signal as the long term delay and white Gaussian

noise as the short term delay that simply model the network delay. Details are included in the Appendix A.

4.4 Packet Delay

Voice packets encounter fixed delays such as encoding/decoding delay, packetization delay, encryption/decryption delay, transmission delay and propagation delay, and dynamic network delay. We chose 10 ms as the encoding/decoding delay when the PCM coding algorithm is used. Packetization delay is selected equal to the time that is needed for collecting samples to form a packet. We neglect DES encryption/decryption delay, transmission delay and propagation delay, since they are rather small compared to other delays.

Once Pareto Distribution random delay is generated, it is assigned to a voice packet as its network delay. Suppose the speech starts at time T_0 , packets are periodically generated and the generating time interval is T_p , encoding delay is D_{enc} and a voice packet is sent immediately when it is generated. Then packet k is transmitted at time $T_s(k)$:

$$T_s(k) = T_0 + kT_p + D_{enc} \quad (\text{Eq.4.9})$$

T_0 is usually set to 0 for simplicity. We are also aware that if the frame-interleaving scheme is adopted, the packet-generating time interval is longer than T_p because of the time needed for collecting interleaved frames.

Let $D(k)$ be the network delay of packet k , then packet k arrives at time $T_r(k)$:

$$T_r(k) = T_s(k) + D(k) \quad (\text{Eq.4.10})$$

If the play out delay is D_{play} , then the scheduled play out time of packet k is:

$$T_{play}(k) = D_{play} + kT_p \quad (\text{Eq.4.11})$$

Then $T_r(k)$ is compared with the scheduled play out time of packet k and the scheduled play out time of packet $k+n+1$, and a drop or save decision is made and the proper buffer is selected depending on the comparison result, as illustrated in Fig. 4.2.

4.5 Buffer Size and Play Out Schedule

Voice packets usually arrive at the receiving end with varying rates although they were generated and sent at a constant rate at the source end. In order to remove the delay variance, an artificial delay is introduced by storing the packets in the jitter buffer. Then voice packets are played out constantly as a scheduled stream. As discussed earlier, 150 – 200 ms is tolerable end-to-end delay for VoIP applications.

The depth of the jitter buffer should be able to store 150 – 200 ms voice packets. For a packet size of 30 ms, it means the capability of holding 5 to 7 packets. If we use the number of packet-streams up to 20, a jitter buffer with capacity of 25 to 27 packets is sufficient. We chose a jitter buffer size of 750 ms and the decryption buffer size of 300 ms.

Because packet delays are randomly distributed, the play out time should be selected in such a manner that the majority of packets can arrive before their scheduled play out time and in the meantime the time-constraint is satisfied as well. Figure 4.8 shows the probability distribution function of Pareto Distribution, given the location parameter set to 5 and with the shape parameter equal to 1.1, 1.3, 1.5, 1.7 and 1.9, respectively. The probability is shown on a logarithmic scale (base 10) for better illustration. In this figure, A and B are two different play out delay values and $A < B$. We clearly see that the probability of packets with delay less than A is smaller than the probability of packets with delay less than B . Therefore, a larger play out time

corresponds to smaller probability of late packets and consequently reduces packet losses, but produces longer end-to-end latency. In addition, it also indicates that given location parameter and play out time, the probability of late packets increases as the shape parameter becomes larger. Because the probability of packet delays less than A is p (i.e. the area left of A and under the PDF curve), then the probability of packet delays longer than A is $1-p$. The larger the shape parameter is, the smaller the value of p is and the larger the value of $1-p$ is.

Figure 4. 8 Pareto Probability Distribution Function

Chapter 5 Simulation Results Analysis

In our simulations, two real speech recordings (#1 and #2) and three types of artificial voice traffic are used. Since VAD is often used in VoIP to reduce bandwidth requirements, we applied the energy-based VAD for the real speech recordings. Lost packets and suppressed silence packets are recovered by inserting 'zero' packets or repeating the previous packets. To avoid a fixed talkspurt (On) / silence (Off) pattern generated from repeating given real speech recordings, random artificial voice traffic with three pairs of commonly used exponential distribution parameters were generated: 352 ms / 650 ms, 227 ms / 596 ms, 1004 ms / 1587 ms. The obtained On/Off patterns are considered VAD results using various VAD algorithms.

5.1 Effect of Number of Packet-Streams

5.1.1 Simulation Results For Real Speech

When using real speech #1, we chose a packet size of 30 ms, play out delay of 150 ms, the number of packet-streams S_n from 1 up to 20, and used a mean delay of 105 ms which is typical for a weekday between 9 am and 10 am [16]. The shape parameter b is chosen as 1.1, 1.4, and 1.9. The jitter buffer size is set at 750 ms and the decryption buffer size is set at 300 ms. Figure 5.1 displays the packet loss rate vs. the number of packet-streams, and Fig. 5.2 displays the average length of packet loss segment vs. the number of packet-streams. By using real speech #2, simulations were repeated as above. The plot in Fig. 5.3 shows the relationship between packet loss rate and number of

packet-streams and the plots in Fig. 5.4 shows the relationship between average length of packet loss segment and number of packet-streams.

Figure 5. 1 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech#1)

Figure 5. 2 Effect of Multiple Packet-Streams on Length of Loss Segment(real speech#1)

Figure 5. 3 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech#2)

Figure 5. 4 Effect of Multiple Packet-Streams on Length of Loss Segment(real speech#2)

Table 5. 1 Simulation Results for Real Speeches

Real speech	Shape parameter b	Packet loss rate (%)		Length of loss segment (ms)		
		Sn=1	Sn=20	Sn=1	Sn=2	Sn=20
#1	1.1	5	3.5	53	33	32
	1.4	10.6	7.5	55	35	33.5
	1.9	11.7	8	57	36	34
#2	1.1	5.6	4	56.5	33	32
	1.4	11.5	8	58	35	33.5
	1.9	13	8.5	59	36	34

Table 5.1 summarizes the simulation results in Fig. 5.1 to 5.4. As the number of packet-streams increases from 1 to 20 and the shape parameter is chosen as 1.1, 1.4 and

1.9, we clearly see that for real speech #1 (Fig. 5.1) the packet loss rate gradually decreases from around 5% to 3.5%, 10.6% to 7.5%, and 11.7% to 8%, respectively; and for real speech #2 (Fig. 5.3) the packet loss rate gradually decreases from around 5.6% to 4%, 11.5% to 8%, and 13% to 8.5%, respectively. These plots indicate that the more packet-streams in use the lower the packet loss rate; and in addition, the larger the shape parameter the higher the packet loss rate. We also observe that for real speech #1 (Fig. 5.2) the length of loss segment decreases from 53 ms to 32 ms, 55 ms to 33.5 ms, and 57 ms to 34 ms. For real speech #2 (Fig. 5.4) the length of loss segment decreases from 56.5 ms to 32 ms, 58 ms to 33.5 ms, and 59 ms to 34 ms. Therefore the more packet-streams the shorter the length of the loss segment, and the larger the shape parameter the longer the length of the loss segment. When two packet-streams are in use, the length of loss segment reduces sharply (to around 33 ~ 36 ms) in these plots and it then decreases very slowly. This means that using two packet-streams will significantly reduce the length of the loss segment and using more packet-streams can make the length of the loss segment gradually approach the minimum length it may have - the size of the packet, in this case 30 ms.

In fact, the played out voice stream obtained sounded more natural compared to the voice stream without using the multiple packet-stream scheme. And when 'silence substitution' and 'repetition' schemes are implemented for lost packet concealment, the second scheme produced more nature voice than that of the first scheme. Refer to the previous discussion about the effects of shape parameter in Sec. 4.5. The simulation results have verified that the larger the shape parameter, the higher the probability of late packets and as a consequence the more packets lost.

5.1.2 Simulation Results for Artificial Speech

In order to use artificial speech for simulations, we chose three common types of artificial speech with pair of talkspurt/silence distribution parameters: type-1 - 352 ms/650 ms, type-2 - 227 ms/596 ms and type-3 - 1004 ms/1587 ms. The other parameters chosen include a packet size of 30 ms, play out delay of 150 ms, the number of packet-streams S_n from 1 up to 20, and mean delay (105 ms) of a typical weekday between 9 am and 10 am. The shape parameter b is chosen as one of 1.1, 1.4, and 1.9. The jitter buffer size is set to 750 ms and the decryption buffer size is set to 300 ms. For each type of artificial speech, we plot the packet loss rate vs. number of packet-streams in Fig. 5.5, 5.7 and 5.9, and the length of loss segment vs. number of packet-streams in Fig. 5.6, 5.8 and 5.10.

Figure 5. 5 Effect of Multiple Packet-Streams on Packet Loss rate (type-1)

Figure 5. 6 Effect of Multiple Packet-Streams on Length of Loss Segment (type-1)

Figure 5. 7 Effect of Multiple Packet-Streams on Packet Loss Rate (type-2)

Figure 5. 8 Effect of Multiple Packet-Streams on Length of Loss Segment (type-2)

Figure 5. 9 Effect of Multiple Packet-Streams on Packet Loss Rate (type-3)

Figure 5. 10 Effect of Multiple Packet-Streams on Length of Loss Segment (type-3)

Table 5. 2 Simulation Results for Three Types of Artificial Speeches

Artificial speech	Shape parameter b	Packet loss rate (%)		Length of loss segment (ms)		
		Sn=1	Sn=20	Sn=1	Sn=2	Sn=20
Type-1	1.1	2.9	1.9	53	32	31
	1.4	6.2	4.1	56.5	35.5	33.5
	1.9	6.7	4.4	57	36	34
Type-2	1.1	2.45	1.7	50	32	32
	1.4	5.15	3.2	54	35	33.5
	1.9	5.4	3.6	55	35	34
Type-3	1.1	3	2.2	53	30	29
	1.4	6.3	4.2	57	34.5	34
	1.9	7.4	4.5	58	35.5	35

Table 5.2 summarizes the simulation results plotted in Fig. 5.5 to 5.10. It is clear that for a given shape parameter b , for each type of artificial speech, larger number of packet-streams S_n corresponds to lower packet loss rate and shorter length of loss segment. For example, using type-1 speech and $b=1.4$, $S_n=1$ and 20, the packet loss rate reduces from 6.2% to 4.1% and the length of loss segment reduces from 56.5 ms to 35 ms. And for a certain S_n , bigger shape parameter b corresponds to higher packet loss rate and longer length of loss segment. For example, using type-2 speech and $S_n=20$, $b=1.1$, 1.4 and 1.9, the packet loss rate increases from 1.7% to 3.6% and the length of loss segment increases from 32 ms to 34 ms. We also observe that when two packet-streams are used, the length of loss segment reduces sharply (to around 30 ~ 36 ms) and then decreases very slowly.

5.1.3 Overall Effect of Multiple Packet-Stream Scheme

Because the traffic on the network is random and unpredictable, we want to investigate the overall effect of the multiple packet-streams scheme on dynamic IP networks. We used a set of mean delay measurements over a 24-hour period in a typical path for our simulations. The packet size is 30 ms, play out delay is 150 ms, the number of packet-streams S_n is from 1 up to 20, and shape parameter b is 1.4. The jitter buffer size is 750 ms and the decryption buffer size is 300 ms. Fig. 5.11 shows the simulation results of the average packet loss rate and Fig. 5.12 shows the simulation results of the average length of packet loss segment.

Figure 5. 11 Overall Effect of Multiple Packet-Streams on Packet Loss Rate

Figure 5. 12 Overall Effect of Multiple Packet-Streams on Length of Loss Segment

As we can see in Fig. 5.11, both the packet loss rate curves decrease as the number of packet-streams increases: from 12.5% to 8% for real speech #1 and from 6.75% to 4.25% for type-1 artificial speech. And in Fig. 5.12, both the lengths of loss segment curves decrease from 58.5 ms to 34 ms for real speech 1 and from 57 ms to 34 ms for type-1 artificial speech; and the length decreases slowly when more than two packet-streams are in use. These plots show that the proposed multiple packet-stream scheme is able to reduce packet loss under dynamic network conditions.

5.2 Effect of Play Out Delay

In order to investigate the effect of play out delay for the multiple packet-stream scheme, for a given play out delay, we denote packet loss rate $PLR(n)$ and the length of loss segment $PLI(n)$ where n is the number of packet-streams and $1 \leq n \leq 20$. Then the reductions of packet loss rate and length of loss segment are defined as the following:

$$PLR_d(n) = PLR(1) - PLR(n) \quad (\text{Eq.5.1})$$

$$PLI_d(n) = PLI(1) - PLI(n) \quad (\text{Eq.5.2})$$

$PLR_d(n)$ and $PLI_d(n)$ are used to indicate the effect of multiple packet-streams. The larger the value of $PLR_d(n)$ and $PLI_d(n)$, the more reduction of packet loss.

By using real speech #1, shape parameter of 1.4, mean delay of 105 ms and play out delay of 150 ms, jitter buffer size of 750 ms and decryption buffer of 300 ms, the simulation and calculation were repeated with play out delay of 100 ms, 120 ms and 150 ms. The results are illustrated in Fig. 5.13 for the reduction of packet loss rate and in Fig. 5.14 for the reduction of length of loss segment. We also performed the same simulations using type-1 artificial speech and the results are shown in Fig. 5.15 for the reduction of packet loss rate and in Fig. 5.16 for the reduction of length of loss segment.

From Fig. 5.13 and 5.15, the plots indicate that for both real speech and artificial speech, shorter play out delay corresponds to larger packet loss rate reduction. For instance, when 12 packet-streams are used and the play out delay is chosen 100 ms, 120 ms and 150 ms, in Fig. 5.13 the reduction of packet loss rate is 4.8%, 3.8%, and 2.9%, respectively; and in Fig. 5.15 the reduction of packet loss rate is 3.3%, 2.7%, and 1.8%, respectively. This means that the multiple packet-streams scheme can be of better use whenever the play out delay is shorter because of the larger packet loss rate reduction. In Fig. 5.14 and 5.16, using play out delay of 100 ms, 120 ms and 150 ms, we also observe the trend that shorter play out delay corresponds to larger reduction of length of loss segment when the number of packet-streams is more than 6. When the number of packet-streams is less than 6, the reduction for longer play out delay is larger than the reduction for shorter play out delay. This is because there are more lost packets with shorter play out delay than with longer play out delay, and thus there are more propagated errors for shorter play out delay than for longer play out delay. When the number of packet-streams is small, the proposed scheme is not able to fully compensate the effect of error propagation for shorter play out delay and thus leads to smaller reduction of length of loss segment.

Figure 5.13 Effect on Packet Loss Rate for Different Play Out Delays (real speech#1)

Figure 5.14 Effect on Length of Loss for Different Play Out Delays (real speech#1)

Figure 5. 15 Effect on Packet Loss Rate for Different Play Out Delays (type-1)

Figure 5. 16 Effect on Length of Loss for Different Play out Delays (type-1)

5.3 Effect of Decryption Buffer and Jitter Buffer

5.3.1 Effect of Decryption Buffer

With decryption buffer size DB from 1 packet to 10 packets, (i.e., buffer size of 30 ms to 300 ms in this case), packet size of 30 ms, play out delay of 150 ms, shape parameter of 1.4, and jitter buffer size JB of 450 ms, we plot the simulation results of the effect of decryption buffer on packet loss rate and length of loss segment. The number of packet-streams S_n is chosen 2, 6, and 10. Figure 5.17 and Figure 5.18 are for real speech sample #1, Fig. 5.19 and Fig. 5.20 are for type-1 artificial speech.

Figure 5. 17 Effect of Decryption Buffer on Packet Loss Rate (real speech#1)

Figure 5. 18 Effect of Decryption Buffer on Length of Loss Segment (real speech#1)

Figure 5. 19 Effect of Decryption Buffer on Packet Loss Rate (type-1)

Figure 5. 20 Effect of Decryption Buffer on Length of Loss Segment (type-1)

Each plot in Fig. 5.17 and Fig. 5.19 gradually reaches a certain packet loss rate and keeps that value, which is the minimum packet loss rate it may reach under those conditions. For example, in Fig. 5.17, given certain jitter buffer size and the number of packet-streams of 2, the plot of packet loss rate is rather flat and kept at 9.7%. The effect of decryption buffer size is not obvious because the number of streams is small and the number of packets that must be stored in buffer for decryption purpose is small as well. As the number of packet-streams becomes larger, the effect of the decryption buffer size increases because a larger decryption buffer is required so as to reach the minimum packet loss rate. We observe that when 6 packet-streams are used, a decryption buffer of more than 3 packets is needed to obtain packet loss rate of 8.3%; when 10 packet-streams

are used, decryption buffer of more than 7 packets is needed to obtain packet loss rate of 7.5%. The simulation results agree to the analysis presented in Sec. 4.1.

Figure 5.18 and Figure 5.20 show that the large number of packet-streams corresponds to shorter length of loss segment and the length approaches a certain value as well. Similar to the plots for packet loss rate, when two packet streams are used, the plot for length of loss segment is relatively flat, but for a larger number of packet-streams, the length decreases until the decryption buffer size is large enough for the number of packet-streams being used.

5.3.2 Effect of Jitter Buffer

To investigate the effect of the jitter buffer, the jitter buffer size JB is chosen from 1 packet to 15 packets, (*i.e.*, buffer size of 30 ms to 450 ms in this case). The shape parameter is chosen as 1.4, the play out delay as 150 ms, and encryption delay as 10 ms. The number of packet-streams S_n is chosen as 2, 6, and 10. Two different decryption buffer sizes DB of 3 packets and 5 packets (*i.e.*, buffer size of 90 ms and 150 ms) were used for each S_n being used. While real speech #1 is used, the simulation results for the effect on packet loss rate are plotted in Fig. 5.21 ($DB=3$) and Fig. 5.22 ($DB=5$); and the simulation results for the effect on length of loss segment are plotted in Fig. 5.23 ($DB=3$) and Fig. 5.24 ($DB=5$). Similarly, the simulation results using type-1 speech are plotted in Fig. 5.25 ($DB=3$) and Fig. 5.26 ($DB=5$) for the effect on packet loss rate, and in Fig. 5.27 ($DB=3$) and Fig. 5.28 ($DB=5$) for the effect on length of loss segment.

Figure 5.21 Effect of Jitter Buffer on Packet Loss Rate (real speech#1, $DB=3$)

Figure 5.22 Effect of Jitter Buffer on Packet Loss Rate (real speech#1, $DB=5$)

Figure 5.23 Effect of Jitter Buffer on Length of Loss Segment (real speech#1, DB=3)

Figure 5.24 Effect of Jitter Buffer on Length of Loss Segment (real speech #1, DB=5)

Figure 5. 25 Effect of Jitter Buffer on Packet Loss Rate (type-1, $DB=3$)

Figure 5. 26 Effect of Jitter Buffer on Packet Loss Rate (type-1, $DB=5$)

Figure 5.27 Effect of Jitter Buffer on Length of Loss Segment (type-1, $DB=3$)

Figure 5.28 Effect of Jitter Buffer on Length of Loss Segment (type-1, $DB=5$)

From these plots, we observe that for given S_n and DB , the packet loss rate reduces as the jitter buffer becomes larger, and when the jitter buffer size is larger than a certain value, the packet loss rate remains constant. Similarly, the length of loss segment reduces as the jitter buffer size increases and remains at a certain value when the jitter buffer size is bigger than certain value. Table 5.3 summarizes Fig. 5.21, 5.22, 5.23 and 5.24, and shows the necessary jitter buffer sizes for the multiple packet-stream scheme to reach the minimum packet loss rate and length of loss segment with those chosen parameters.

Table 5. 3 Required Jitter Buffer Size for Minimum Packet Loss

DB (packet)	S_n	JB (packet)	Packet loss rate (%)	JB (packet)	Length of loss segment (ms)
3	2	3	12	2	36
	6	7	10	6	35
	10	11	8	10	34.5
5	2	1	11.5	1	36
	6	5	10	5	34.5
	10	9	9	9	34

In the case when $DB=3$ packets, and S_n chosen as 2, 6 and 10, the jitter buffer size should be at least 3, 7 and 11 packets respectively to approach its minimum packet loss rate and the jitter buffer size should be 2, 6 and 10 packets, respectively, to approach its minimum length of loss segment. Therefore in order to obtain both minimum packet loss rate and length of loss segment for the given S_n (2,6,10) and DB (3 packets), the jitter buffer should be chosen 3, 7 and 11 packets. Similarly, in the case $DB=5$ packets, $S_n=2, 6$ and 10, the jitter buffer size should be at least 1, 5 and 9 packets, respectively, to

approach its minimum packet loss rate and length of loss segment with the chosen parameters. From Table 5.1, we clearly see that when S_n increases, JB should increase so as to make the best use of the multiple packet-stream scheme (i.e., approach the minimum packet loss rate and length of loss segment). We note that the increase in S_n is equal to the increase required for JB . If 4 more packet-streams are included (e.g. S_n increases from 2 to 6 or from 6 to 10), a large jitter buffer that can hold 4 more packets is needed (e.g., in the first case JB from 3 to 7 packets or from 7 to 11 packets and in the second case JB from 1 to 5 packets or from 5 to 9 packets).

In our buffering algorithm those packets that are dropped due to the overflow of the jitter buffer can be stored in the decryption buffer when the decryption buffer is not full. According to Fig. 5.21 and 5.23 in the first case ($DB=3$) when $S_n=2$, the decryption buffer can't hold all the overflowed packets if $JB=1$ or 2 packets. So a jitter buffer size of more than 3 packets is needed. According to Fig. 5.22 and 5.24 in the second case ($DB=5$) when $S_n=2$, it seems that the jitter buffer size doesn't affect the packet loss rate and length of loss segment because the decryption buffer is large enough to store the overflowed packets even when $JB=1$ packet. From these two cases, we see that the larger the decryption buffer, the smaller the required jitter buffer. So the decryption buffer can be used as a backup buffer if the jitter buffer has overflowed.

We also analyzed the plots for type-1 speech in Fig. 5.25, 5.26, 5.27 and 5.28 in the same way as we did to Fig. 5.21, 5.22, 5.23 and 5.24, and obtained the similar results.

5.4 Effect of Frame-Interleaving

We also investigated the multiple packet-stream scheme with frame-interleaving

implemented. We grouped every 80 PCM samples as a “frame”, then the size of the frame is 10 ms and the packet size is chosen 30 ms (3 frames/packet). We used a play out delay of 150 ms, shape parameter of 1.4, mean delay of 105ms, and the frame-interleaved space of 3. The jitter buffer is 750 ms and the decryption buffer is 300 ms. Figure 5.29 and Figure 5.30 plot the simulation results for packet loss rate and length of loss segment, respectively.

In Fig. 5.29 we clearly see that as the number of packet-streams increases from 1 to 20, the packet loss rate gradually decreases from 11.8% to 8% for real speech #1, from 6% to 4% for type-1 speech, from 5% to 3.2% for type-2 speech and 6.5% to 5% for type-3 speech; and in Fig. 5.30, the length of loss segment also reduces from 17.2 ms to 11.3 ms for real speech #1, from 17.5 ms to 11.2 ms for type-1 speech, from 17.8 ms to 11.2 ms for type-2 speech and from 18.2 ms to 11.1 ms for type-3 speech. Since the length of loss segment is very close to the size of a frame (i.e., 10 ms), which is the minimum length of loss segment the frame-interleaving scheme can achieve, the simulation results indicate that the two schemes can be combined to provide better QoS.

Figure 5. 29 Effect on Packet Loss Rate with Frame-Interleaving Implementation

Figure 5. 30 Effect on Length of Loss Segment with Frame-Interleaving Implementation

Chapter 6 Conclusions

Packet delay and packet loss are common impairments of voice services over IP networks. Since more and more telephone conversations are transmitted over IP networks, privacy is of great concern to the users, especially when sensitive information is delivered. In order to provide acceptable QoS for secure VoIP, certain constraints must be taken into account while implementing cryptographic systems.

In this thesis, we investigated the stringent time requirements of encrypted VoIP and analyzed the error propagation characteristic (i.e. packet decryption depending on the availability of previous packet) of CBC mode and CFB mode of the DES encryption algorithm and presented a multiple packet-stream scheme where the effects of packet latency loss are reduced for CBC and CFB encryption modes. The proposed solution is able to relax the time constraints linearly by the number of packet-streams for the packet decryption, which results in a reduction in overall packet loss due to latency.

In our simulations, a voice packet size of 30 ms in PCM encoding is chosen; both real speech recordings and exponentially distributed artificial speeches are used. Energy based VAD and a frame-interleaving scheme are implemented and random packet delays are modeled and generated as Pareto Distribution. Our simulation results clearly indicate that the proposed multiple packet-stream scheme can not only reduce packet loss rate but also shorten the length of loss segment, - the more packet-streams being used, the greater reduction of packet loss. (Another network delay model is also used to generate bursty delay, see detail simulation results in the Appendix A)

The play out delay affects the effectiveness of the multiple packet-stream scheme,

in that the voice stream with shorter play out delay benefits more from the proposed solution than the voice with longer play out delay.

A larger decryption buffer and jitter buffer are necessary when more packet-streams are included. In our implementation, the decryption buffer can not only store late packets for decryption purposes, but also acts as a backup buffer when the jitter buffer overflows. When the buffer sizes are properly chosen, the multiple packet-stream scheme can produce the best results with those chosen parameters.

When the multiple packet-stream scheme is implemented along with a frame-interleaving scheme, its feature of reducing packet loss still stands well. And the frame-interleaving scheme is also successful in splitting loss segment. Thus VoIP can make use of the benefits of both schemes.

We also point out that the proposed multiple packet-stream scheme allows using a different encryption key in each packet-stream so as to better protect message privacy. When one key is broken, the more encryption keys are used in the multiple packet-streams, the smaller portion of a message is leaked, and in addition the distance between two disclosed segment is greater and the disclosed content is harder to understand. The introduced delay for key management is rather small compared to other delays the voice packets may encounter; and the extra resource consumption is also very small, compare to the additional resource required by using longer encryption key.

During the research, we found that the error-propagation-free characteristic of OFB mode and CTR mode are very suitable for real-time applications. These modes could be potential candidates in encrypted VoIP. The main concerns of using OFB mode and CTR mode are:

- How to detect the lost synchronization between the decryption input and the arriving packets
- How to detect the lost synchronization in a timely manner so that the errors are limited within QoS constraints
- How to re-synchronize the decryption input and the arriving packets (periodically or on-demand)
- How to reduce the overhead of re-synchronization (in terms of delay and bandwidth utilization)
- Compare the performance to that of the proposed scheme

Appendix A: Additional Simulation Results Analysis

To model the network delay that has bursty characteristic, we chose white Gaussian noise as the short term delay on top of a sine signal as the long term delay. We performed some initial simulations to investigate the effectiveness of the proposed scheme under these conditions.

The fixed network delay is set to 90 ms, and the amplitude and period of the sine signal are set to 30 ms and 2.5 second, respectively. The mean of the white Gaussian noise is chosen 0 and the standard deviation is 30 ms.

Fig. A.1 shows the delays used in our simulations.

Figure A. 1 Delays Generated by A Bursty Network Delay Model

The simulation environment is similar to that is used for Pareto distribution. We use a packet size of 30 ms, play out delay of 150 ms, the encoding delay of 10 ms. Figure A.2 and A.3 show the effect of the number of packet streams on packet loss using real speech #1. Figure A.4 and A.5 show the effect using type-1 artificial speech and figure A.6 and A.7 show the effect using type-2 artificial speech.

Figure A. 2 Effect of Multiple Packet-Streams on Packet Loss Rate (real speech #1)

In figure A.2, the plot shows that the packet loss rate decreases from 20.4% to 17.6% when S_n reaches 5 but then tends to increase slightly as S_n increases up to 20. Figure A.4 and A.6 have similar results; the only difference is that the packet loss rate starts increasing at a different value of S_n . But it is clear that the increase is within 1.5% and the packet loss rate with more packet-streams is always lower than using single packet-stream. The packet loss

rate increase with more packet-streams may be caused by the bursty packet loss. For example, if 3 packet-streams are included and there are 9 packet losses in a row at a certain time, according to the error propagation characteristic, 3 more packet losses are produced. So a total of 12 packets are lost. In case the number of packet-streams is increased to 9, for this 9-packet long loss segment, another 9 packets are lost due to error propagation. So a total of 18 packets are lost. Therefore the packet loss rate may increase with more packet-streams in some circumstance. Figure A.3, A.5 and A.7 show the effect of multiple packet streams on the length of loss segment. Similar to the analysis for packet loss rate, the length of loss segment decreases as S_n increases to a certain value but it is not in a monotonic manner and can sometimes result in slightly longer packet loss segments for larger number of packet-streams. For example, in figure A.3, the loss segment reduces from 61 ms to 49 ms when S_n reaches 4, then the value is up and down in a small range with more packet-streams.

So the effect of the multiple packet-streams on packet loss is not as consistent as the effect by using Pareto distributed delay. The burstiness of lost packet may affect the effectiveness of multiple packet-streams.

Figure A. 3 Effect of Multiple Packet-Streams on Length of Loss Segment (real speech#1)

Figure A. 4 Effect of Multiple Packet-Streams on Packet Loss Rate (type-1)

Figure A. 5 Effect of Multiple Packet-Streams on Length of Loss Segment (type-1)

Figure A. 6 Effect of Multiple Packet-Streams on Packet Loss Rate (type-2)

Figure A. 7 Effect of Multiple Packet-Streams on Length of Loss Segment (type-2)

REFERENCES

- [1] S. Wang, L. Nieto and E. Zielinski, "A study of IP network impairments impact on media gateway performance," *GLOBECOM '01. IEEE*, vol. 4, pp. 2596 - 2600.
- [2] W. Kampichler and K.M. Goeschka, "Measuring voice readiness of local area networks," *GLOBECOM '01. IEEE*, vol. 4, pp. 2501 -2505.
- [3] E. Weiss, "Security concerns with VOIP," Aug. 20, 2001, http://rr.sans.org/VOIP/sec_concerns.php
- [4] J. Postel, "Transmission Control Protocol," IETF RFC 793, Sept. 1981.
- [5] J. Postel, "User Datagram Protocol," IETF RFC 768, Aug. 1980.
- [6] H. Schilzrinne et al., "RTP: A transport protocol for real-time Applications," IETF RFC 1889, Jan. 1996.
- [7] ITU-T H.323, "Packet based multimedia communications systems," Feb. 1998.
- [8] J. Sohn, N. S. Kim, W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, issue. 1, pp. 1-3, Jan 1999.
- [9] J. Rosenberg et al. "SIP: Session initiation protocol," IETF RFC 2543, June 2002.
- [10] ITU-T H.235 version 2 "Security and encryption for H-Series multimedia terminals," Nov. 2000.
- [11] ITU Recommendation G.711, "Pulse Code Modulation (PCM) of Voice Frequencies," 1988.

- [12] V. Matic, A. Bazant, M. Kos, "Predictive playout delay adaptation for Voice over Internet," *MELECON 2000. 10th, Mediterranean*, vol. 1, pp. 348 -351.
- [13] J. Jo, G. Hwang, H. Yang, "Characteristics of QoS parameters for VoIP in the short-haul Internet," *Info-tech and Info-net, Proc. ICII*, vol. 2, pp. 498 -502, Beijing 2001.
- [14] P. Mehta, S. Udani, "Voice over IP," *IEEE Potentials*, vol. 20, issue. 4, pp. 36 - 40 Oct-Nov 2001.
- [15] ITU-T Recommendation G.114, "One-way Transmission Time," 1996.
- [16] T.J. Kostas, M.S. Borella, I. Sidhu, G.M. Schuster, J. Grabiec, J. Mahler, "Real-time voice over packet-switched networks," *IEEE Network*, vol. 12, issue. 1, pp. 18-27, Jan.-Feb. 1998.
- [17] R. Ramaswamy, "Design of a secure packet voice communication system in wide area networks," *Security Technology, Crime Countermeasures, Proc. IEEE Int. Carnahan Conf.* 1990.
- [18] G. Thomsen, Y. Jani, "Internet telephony: going like crazy," *IEEE Spectrum*, vol. 37, issue. 5, pp. 52-58, May 2000.
- [19] B. Vlaovic, Z. Brezocnik, "Packet based telephony," *EUROCON'2001 Trends in Communications, Int. Conf.* vol. 1, pp. 210-213.
- [20] L. Zheng, L. Zhang, D. Xu, "Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP)" *ICC IEEE Int. Conf.*, vol. 1, pp. 122 - 126, Jun 2001.
- [21] M.J. Karam, F.A. Tobagi, "Analysis of the delay and jitter of voice traffic over the Internet," *INFOCOM. Proc. IEEE*, vol. 2, pp. 824 -833, 2001.

- [22] N. Laoutaris, I. Stavrakakis, "Intrastream synchronization for continuous media streams: a survey of playout schedulers," *IEEE Network*, vol. 16, issue. 3, pp. 30 – 40, May-June 2002. Page(s): 30 –40.
- [23] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," *INFOCOM, Networking for Global Communications.*, 13th Proc. IEEE, vol. 2, pp. 680 –688, 1994.
- [24] H. Melvin, L. Murphy, "Time synchronization for VoIP quality of service," *IEEE Internet Computing*, vol. 6, issue. 3, pp. 57 –63, May-June 2002.
- [25] C. Padhye, K.J. Christensen, W. Moreno, "A new adaptive FEC loss control algorithm for voice over IP applications," *IPCCC Proc. IEEE Int. Conf.* pp. 307 –313, Feb 2000.
- [26] A. Sangwan, M.C. Chiranth, H.S Jamadagni, R. Sah, R.V. Prasae, V. Gaurav, "VAD techniques for real-time speech transmission on the internet," *High Speed Networks and Multimedia Communications 5th IEEE Int. Conf.* pp. 46 – 50.
- [27] Y.D. Cho, K. Al-Naimi, A. Kondo, "Mixed decision-based noise adaptation for speech enhancement," *Electronics Letters*, vol. 37, issue. 8, pp. 540 –542, Apr 2001.
- [28] M.S. Borella, D. Swider, S. Uludag, G.B. Brewster, "Internet packet loss: measurement and implications for end-to-end QoS," *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing.*, Proc. ICPP Workshops, pp. 3 –12, 1998.
- [29] C. Perkins, O. Hodson, V. Hardman, "A survey of packet loss recovery

- techniques for streaming audio," *IEEE Network*, vol. 12, issue. 5, pp. 40-48 Sept.-Oct. 1998.
- [30] B.W. Wah, S. Xiao, L. Dong, "A survey of error-concealment schemes for real-time audio and video transmissions over the Internet," *Multimedia Software Engineering, Proc. Int. Symposium* pp. 17-24, 2000.
- [31] D.W. Davies and W.L. Price, *Security for Computer Networks*, 2nd Edition, John Wiley & Sons 1989.
- [32] N. Kovblitz, *A Course in Number Theory and Cryptography*, 2nd Edition, Springer-Verlag 1994.
- [33] J. Calpe, J.R. Magdalena, J.F. Guerrero, J.V. Frances, "Toll-quality digital secraphone," *MELECON*, vol. 3, pp. 1714 -1717, Mediterranean, 13-16 May 1996.
- [34] I.B. Fernandez, W.V. Subbarao, "Encryption based security for ISDN communication: technique & application," *IEEE* pp. 70 -72 1994.
- [35] C. Boyd, "Modern data encryption," *Electronics & Communication Engineering Journal*, vol. 5, issue. 5, pp. 271 -278, Oct 1993.
- [36] I.B. Fernandez, W.V. Subbarao, "Encryption based security for ISDN communication: technique and application," *Southeastcon 'Creative Technology Transfer - A Global Affair'. Proc. EEE*, pp. 70 -72, 10-13 Apr 1994.
- [37] DRAFT ITU-T Recommendation H.Secure: "Security and encryption for H Series (H.323 and other H.245 based) multimedia terminals," Appendix A: encryption algorithm criteria, Sept. 1996.

- [38] F. Simonds, *Network Security (data and voice communication)*, McGraw-Hill series on computer communications (1996).
- [39] RSA laboratories, "Answers to frequently asked questions about today's cryptography," <ftp://ftp.funet.fi/pub/crypt/cryptography/asymmetric/rsa/rsa-faq-1.txt>
- [40] S. Jee, P. Montague, "A secure DES implementation for real-time embedded applications," *The third Int Conf. on Knowledge-based Intelligent Information Engineering System*, Adelaide, Australia, Aug. 31- Sept. 1, 1999.
- [41] "Data Encryption Standard (FIPS PUB 46-3 reaffirmed)," NIST Oct. 1999.
- [42] M. Dworkin, "Recommendation for block cipher modes of operation: methods and techniques," NIST Special Publication 800-38A, 2001.
- [43] J. Daemen, V. Rijmen, "Advance Encryption Standard (FIPS PUB 197)," NIST May. 2002.
- [44] D.A. Carts, "A review of the Diffie-Hellman algorithm and its use in secure internet protocols," Nov. 5, 2001, <http://rr.sans.org/encryption/algorithm.php>
- [45] G. Troullinos, "A software based approach to secure voice applications," *ICECS96, Proc. of the Third IEEE Int. Conf.*, vol.1, pp. 176 –182.
- [46] S. Jee; P. Montague, "A secure DES implementation for real-time embedded applications," *Knowledge-Based Intelligent Information Engineering Systems, 1999. Third Int. Conf.* pp. 496 –500, Dec 1999.
- [47] M. Marjalaakso, "Security requirements and constraints of VoIP," <http://www.hut.fi/~mmarjala/voip>
- [48] W. Jiang; H. Schulzrinne, "Analysis of on-off patterns in VoIP and their effect

- on voice traffic aggregation," *Computer Communications and Networks Proc. Ninth Int. Conf.* pp. 82–87 2000.
- [49] S. Deng, "Traffic characteristics of packet voice," *ICC 95 Seattle, Gateway to Globalization, IEEE Int. Conf.* vol. 3, pp. 1369, Jun 1995.
- [50] H. Lee, C. Un, "A study of on-off characteristics of conversational speech," *Communications, IEEE Trans.* vol. 34, issue: 6, pp. 630–637, Jun 1986.
- [51] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," Technical report #: MS-CIS_92-83/DSL-12), Computer and Information Science Depart., Univ. of Pennsylvania, Philadelphia, PA, Dec 7 1992.
<http://citeseer.nj.nec.com/516363.html>
- [52] V. Paxson, S. Floyd, "Wide area traffic: the failure of Poisson modeling," *Networking IEEE/ACM Trans.* vol. 3, issue. 3, pp. 226–244, June 1995.
- [53] Q. Li, D.L. Mills, "On the long-range dependence of packet round-trip delays in Internet," *ICC IEEE Int. Conf.* vol. 2, pp. 1185–1191, 7-11 Jun 1998.
- [54] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *Networking, IEEE/ACM Trans.* vol. 2, issue. 1, pp. 1–15, Feb 1994.
- [55] F. Yegenoglu, F. Faris, O. Qadan, "A model for representing wide area Internet packet behavior," *PCCC. Proc. IEEE Int. Conf.* pp. 167–173, Feb 2000.
- [56] M.E. Crovella, A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *Networking, IEEE/ACM Trans.* vol. 5, issue. 6, pp. 835–846, Dec. 1997.

- [57] K. Fujimoto, S. Ata, M. Murata, "Playout control for streaming applications by statistical delay analysis," *ICC IEEE Int. Conf.* vol. 8, pp. 2337–2342, 2001.