A SECURITY MODEL OF VOICE EAVESDROPPING PROTECTION OVER DIGITAL NETWORKS

SATHAPORN KASSUVAN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(COMPUTER SCIENCE)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2006**

ISBN 974-04-7458-6

Thesis
Entitled

A SECURITY MODEL OF VOICE EAVESDROPPING PROTECTION OVER DIGITAL NETWORKS

.....
Mr. Sathaporn Kassuvan
Candidate

.....
Assoc. Prof. Supachai Tangwongsan, Ph.D.
Major-Advisor

.....
Asst. Prof. Chomtip Pornpanomchai, D.Tech.Sc.
Co-Advisor

.....
Lect. Supatana Auethavekiat, Ph.D.
Co-Advisor

.....
Prof. Dr. M.R. Jisnuson Svasti, Ph.D.
Dean
Faculty of Graduate Studies

.....
Assoc. Prof. Supachai Tangwongsan, Ph.D.
Chair
Master of Science Programme in
Computer Science
Faculty of Science

Thesis
Entitled

A SECURITY MODEL OF VOICE EAVESDROPPING PROTECTION OVER DIGITAL NETWORKS

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science (Computer Science)
on
17 May, 2006

.....
Mr. Sathaporn Kassuvan
Candidate

.....
Assoc. Prof. Supachai Tangwongsan, Ph.D.
Chair

.....
Asst. Prof. Chomtip Pornpanomchai, D.Tech.Sc.
Member

.....
Lect. Supatana Auethavekiat, Ph.D.
Member

.....
Assoc. Prof. Chinda Achariyakul, Ph.D.
Member

.....
Prof. Dr. M.R. Jisnuson Svasti, Ph.D.
Dean
Faculty of Graduate Studies
Mahidol University

.....
Prof. Amaret Bhumiratana, Ph.D.
Dean
Faculty of Science
Mahidol University

ACKNOWLEDGEMENT

The success of this thesis can be attributed to the extensive support and assistance from my thesis advisor, Assoc. Prof. Dr. Supachai Tangwongsan. I deeply thank him for his valuable suggestions, comments and encouragement throughout this thesis, without him the thesis would have never been completed.

I am deeply grateful to Asst. Prof. Dr. Chomtip Pornpanomchai, Dr. Supatana Auethavekiat, and Assoc. Prof. Dr. Chinda Achariyakul, my committees, for their kindness in providing valuable suggestions and comments for improvement during the thesis defense.

I would like to acknowledge all the instructors and staff of the Department of Computer Science for their valuable advice and kind support. Many thanks are given to all my friends in the Computer Science class for their kind support and warm relationship at Mahidol University.

Finally, I would like to express my utmost gratitude to my family for their entire care, love and moral support. The usefulness of this thesis, I dedicate to my beloved father who passed away, my mother and all the teachers who have taught me since my childhood.

Above all, thank you for a law of nature that makes human being have a sense of learning and improving.

Sathaporn Kassuvan

A SECURITY MODEL OF VOICE EAVESDROPPING PROTECTION OVER DIGITAL NETWORKS.

SATHAPORN KASSUVAN 4437184 SCCS/M

M.Sc.(COMPUTER SCIENCE)

THESIS ADVISOR : SUPACHAI TANGWONGSAN, Ph.D., CHOMTIP PORNPANOMCHAI, D.Tech.Sc., SUPATANA AUETHAVEKIAT, Ph.D.

ABSTRACT

The purpose of this research is to develop a security model for voice eavesdropping protection over digital networks. The proposed model provides an encryption scheme and a personal secret key exchange between communicating parties, a so-called voice data transformation system, resulting in a real-privacy conversation.

Operation of this system comprises two main processes. The first one is the personal secret key exchange to allow use of a key in the encryption process during the conversation. The other one is the encryption of G.729, G.723.1, G.726 or AMR encoded voice data in the conversation. In doing so, users can choose the option of encryption by themselves, that is, either all frame or partial frame encryption. In partial frame encryption or 1-to-M block encryption, the first block of each packet is encrypted while the remaining M-1 blocks are left unchanged, alternating like this until the end of the conversation. In this process, M is the number of blocks in each group of packets to be encrypted, possibly 1, 2, 3, 4, or 5, giving those who wish to tap the encrypted voice or apply a method of cryptanalysis to the conversation, a difficult task.

The system prototype was implemented and tested based on its conceptual design. The results of the implementation indicate that the system can perform its function accurately as designed. Even in the case of 20% of the whole data being encrypted, the system could still provide speech security. Moreover, partial frame encryption would take less time than all frame encryption, which the experimental result yields quite satisfactorily. In this regard, the proposed system is suitable for effective use in voice eavesdropping protection over digital networks, without the requirement to change presently existing network systems. Also, it would be possible to extend this system to mobile phone networks or Internet networks for the sake of business purposes.

KEY WORDS: SECURITY MODEL / VOICE EAVESDROPPING PROTECTION / DIGITAL NETWORKS

65 P. ISBN 974-04-7458-6

โมเดลความปลอดภัยของการป้องกันการดักฟังบนเครือข่ายดิจิตอล (A SECURITY MODEL OF VOICE EAVESDROPPING PROTECTION OVER DIGITAL NETWORKS)

สถาพร กาศสุวรรณ 4437184 SCCS/M

วท.ม. (วิทยาการคอมพิวเตอร์)

คณะกรรมการควบคุมวิทยานิพนธ์ : ศุภชัย ตั้งวงศ์ศานต์, Ph.D., ชมทิพ พรพนมชัย, D.Tech.Sc., สุพัฒนา เอื้อทวีเกียรติ, Ph.D.

บทคัดย่อ

การศึกษาครั้งนี้เป็นการวิจัยเพื่อพัฒนา โมเดลความปลอดภัยของการป้องกันการดักฟังบนเครือข่ายดิจิตอล โดยการสร้างวิธีการเข้ารหัสและการแลกเปลี่ยนคีย์ (key) เป็นการส่วนตัวระหว่างคู่สนทนา หรือเรียกว่าระบบการแปลงรูปแบบข้อมูลเสียง (Voice Data Transformation System) ซึ่งจะทำให้เกิดความเป็นส่วนตัวในการสนทนาอย่างแท้จริง

การทำงานของระบบดังกล่าวประกอบด้วยสองขั้นตอนหลัก ดังนี้ ขั้นตอนแรกคือการแลกเปลี่ยนคีย์เป็นการส่วนตัว เพื่อที่จะได้นำคีย์นั้นไปใช้ในการเข้ารหัสในระหว่างการสนทนาและขั้นตอนที่สองคือการเข้ารหัสข้อมูลเสียงระหว่างการสนทนาที่ถูกบีบอัดข้อมูลด้วยมาตรฐาน G.729, G.723.1, G.726 หรือ AMR โดยผู้ใช้งานสามารถที่จะเลือกทางเลือก (option) ในการเข้ารหัสได้ด้วยตัวเองว่าจะใช้การเข้ารหัสข้อมูลทั้งหมดหรือการเข้ารหัสข้อมูลแต่เพียงบางส่วน สำหรับในกรณีการเข้ารหัสเพียงบางส่วนหรือการเข้ารหัสแบบ 1-to-M block นั้นเป็นการเข้ารหัสที่บล็อก (block) แรกของข้อมูลและเว้นไว้ไม่เข้ารหัสเป็นจำนวน M-1 บล็อกถัดไปสลับอย่างนี้ไปจนจบการสนทนา ทั้งนี้ M คือจำนวนบล็อกในแต่ละกลุ่ม (group) ของข้อมูล (packets) ที่นำมาเข้ารหัส ซึ่งมีค่าที่เป็นไปได้เท่ากับ 1, 2, 3, 4 หรือ 5 โดยวิธีการดังกล่าวจะยากสำหรับผู้ดักฟังที่พยายามจะเปิดเผยข้อมูลเสียงของผู้พูดหรือคีย์ลับ (cryptanalysis)

จากนั้นเราได้ทำการทดลองปฏิบัติและทดสอบต้นแบบของระบบ (system prototype) บนพื้นฐานของการออกแบบเชิงแนวคิด (conceptual design) ซึ่งผลของการทดลองปฏิบัตินั้นพบว่าระบบสามารถทำงานได้อย่างถูกต้องตามที่ได้ออกแบบไว้ และแม้ว่าในกรณีของการเข้ารหัสที่ 20 เปอร์เซ็นต์ของข้อมูลทั้งหมด ระบบยังคงมีความปลอดภัยของข้อมูลเสียงอยู่ นอกจากนี้แล้วการเข้ารหัสเพียงบางส่วนยังใช้เวลาในการเข้ารหัสน้อยกว่าการเข้ารหัสข้อมูลทั้งหมดซึ่งถือได้ว่าผลการทดลองที่ได้เป็นที่น่าพอใจ ทั้งนี้ระบบที่ได้นำเสนอนี้เหมาะสมที่จะนำไปใช้เพื่อป้องกันการดักฟังเสียงบนเครือข่ายดิจิตอลได้อย่างมีประสิทธิภาพโดยไม่ต้องมีการปรับเปลี่ยนระบบเครือข่ายเดิม และยังเป็นไปได้ที่จะนำระบบนี้ไปประยุกต์ใช้จริงกับเครือข่ายโทรศัพท์เคลื่อนที่หรือเครือข่ายอินเตอร์เน็ตเพื่อให้เกิดโอกาสทางธุรกิจอีกด้วย

65 หน้า. ISBN 974-04-7458-6

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

LIST OF TABLES

LIST OF TABLES (Cont.)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

CHAPTER I
INTRODUCTION

This research presents a model of voice eavesdropping protection system over digital networks with a highly personalized security. In particular, the two parties of conversation could set their own encryption keys and rules, other than solely from their network service providers, in which we could expect a better level of performance in security.

As we enter the second millennium, we experience one of the most important changes in our lives, the move to a knowledge based society. Almost everything will be changed at home, in school, at work, in the government. Some changes are already here and they are spreading around the globe. Others are just beginning. Digital networking system is one of key infrastructures in the knowledge based society as it provides information and knowledge exchanges in forms of voice, data, image or video.

Voice on the digital network is sliced up into small packets, each carrying its own copy of the destination address. The packets travel individually to their destination, not necessarily over the same route, and are reassembled in proper sequence when they arrive. The first version of digital voice was packed in 64 kb/s channel; however, the present versions are in the standard of G.729 and G.723.1 with 6.3 and 5.3 kb/s, respectively.

Security is required for reliable digital voice development. Unfortunately, securing voice over digital networks such as VoIP (Voice over Internet Protocol) is more difficult than securing traditional, circuit-switched voice networks. Now, voice is vulnerable to worms, viruses, denial of service and eavesdropping if not properly handled.

The present work is concerned with the problem of voice eavesdropping in particular and how to develop a model of protection system with a highly personalized security. The following, a working principle of the proposed model is described.

Briefly speaking, the speaker's voice data are encoded according to the standard coding scheme, and then encrypted by his/her personal key before sending over the network. And a receiving device will decrypt the arrival packets with another personal key accordingly, and decode the resultant packets back into its original voices. As for a key used in the encryption, and another key used in the decryption processes are determined by the two conversation parties, it could be seen that the proposed model is highly personalized for security protection. Furthermore, the speakers in the conversation parties could set their encryption and decryption rules, for example, encrypt 20% or 25% of the whole voice packets. By this technique, it is rather difficult for eavesdroppers to discover the speaker's voice or the secret keys by applying the so-called cryptanalysis. This is because the eavesdroppers do not know which parts are encrypted (discussed later). In addition, partial encryption takes less time than full encryption.

In this research, an overall experiment comprises three main sections. Section I provides setting up key exchange to use in speech encryption and decryption. Section II, the encoded voice data are encrypted at 100%, 50%, 33%, 16%, 9% of the whole data to evaluate that which encryption scheme takes less time and keeps speech security. And in Section III, an appropriate encryption scheme derived from Section II is tested together with key exchange by simulating an eavesdropping circumstance.

Based on experimental results, it has been found that the proposed system can perform its function accurately, corresponding to system design objectives with the effectiveness of voice eavesdropping protection. With the success of our prototype implementation, we hope that the present model may lead to any further development in the future.

The contents of this research are organized to seven chapters. Besides this chapter, the rest of them are as follows.

Chapter II Problem statements. We describe the problem statements, briefings of related literature surveyed as well as research objective and scope.

Chapter III Conceptual design. We describe the conceptual design of voice eavesdropping protection called the voice data transformation model for the identified problems.

Chapter IV System design. We present the system design by the structure chart of the system as well as the description of each functional process.

Chapter V System implementation. We describe how to implement the designed system.

Chapter VI Experimental results. We perform step-by-step experiments as well as recording the experimental results for further comparison.

Chapter VII Discussion and conclusion. This chapter covers the discussion of the experimental results mentioned in the following issues: accuracy, performance, application, and program complexity. In the conclusion, we evaluate whether the defined research objective has been fulfilled.

CHAPTER II
PROBLEM STATEMENTS

Nowadays, we are in the boundless-communication age. In other words, people from different parts of the world can freely contact each other. As a result, network communications technology becomes another choice that plays a crucial role in our today life, for instance, the mobile phones or the Internet network such as VoIP. All of these communication forms provide us with rapid and immediate results. However, the voice data transmitted over the network should be seriously considered in terms of security. Therefore, to achieve the highest efficiency, we should have a proper voice eavesdropping protection to preserve the privacy of communicating parties. Moreover, we can apply this protection scheme over the network to other fields for the sake of commercial use or military action as well.

2.1 Motivations

Today, people need more communications, making the network communications like wireless links such as the mobile phones, or communications via the Internet such as VoIP, becomes a part of new-age life.

However, the voice eavesdropping protection over the network system today has been further developed to achieve high effectiveness. For instance, in the mobile phone system, it is not quite the real privacy. This is because it does not provide an end-to-end security. So, the voice data within the network can be captured, resulting in the intercepted conversation. Other than that, users cannot control the encryption keys by themselves. As a result, many severe problems may happen, for instance, the communicating parties will lose their privacy in the conversation situation, or else they may lose business benefits, or, at the worst, military secrets.

Therefore, in this research, we aim at finding out an approach to use in preventing voice eavesdropping as mentioned earlier.

2.2 Literature Survey

This section concludes the related literature of eavesdropping protection over the mobile phones or the Internet network. In order to develop the eavesdropping protection over the network, the existing approaches should be mentioned. That is,

1. Perception-based partial encryption
2. Hierarchical Data Security Protection
3. Secure Real-time Transport Protocol
4. Security for GSM network
5. End-to-end security for GSM users

And the details of them are described as the following.

2.2.1 Perception-based partial encryption

In this section, the existing literature of perception-based partial encryption [1] is presented. This model is used in encryption to prevent mobile multimedia applications from eavesdropping during the conversation. By this model, the bit streams of G.729 [2] encoded voice data are partially encrypted. And the data are divided into two segments, one to be encrypted and the other to be left unencrypted as shown in Figure 2.1.

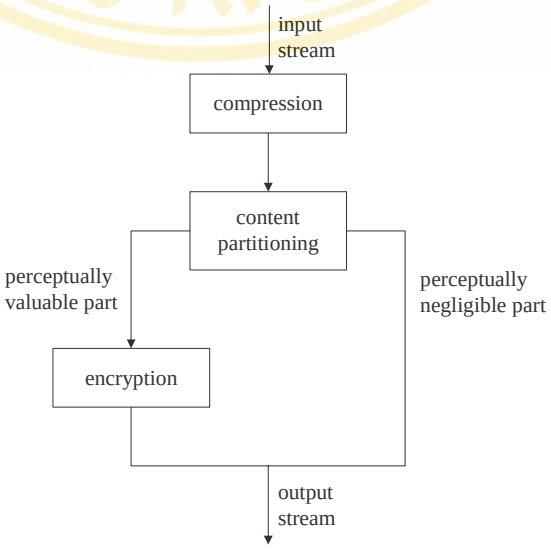

Figure 2.1 Partial encryption scheme

In the case of the encrypted data, the exact position of bits of the voice data to be encrypted is specified as illustrated in Figure 2.2. At the same time, the definition of bit allocation for the ITU-T G.729 standard is shown in Table 2.1.

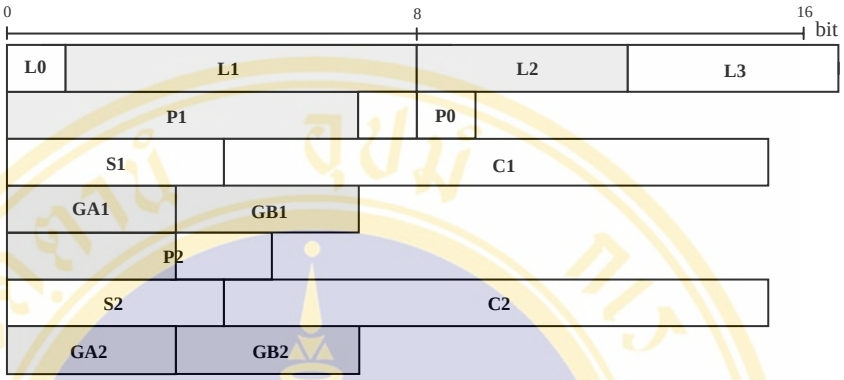

Figure 2.2 Partial encryption schemes for G.729: bit allocations are shown, MSB (the most significant bit) to LSB (the least significant bit) moving from left to right. Bits subject to encryption are shown in gray.

Table 2.1 Bit allocation for ITU-T G.729

Symbol	Description	Bits
L0	Switched MA predictor index of LSP quantizer	1
L1	First stage vector of LSP quantizer	7
L2	Second stage lower vector of LSP quantizer	5
L3	Second stage higher vector of LSP quantizer	5
P1	Pitch delay 1st subframe	8
P0	Parity bit for pitch delay	1
S1	Signs of fixed-codebook pulses 1st subframe	4
C1	Fixed codebook 1st subframe	13
GA1	Gain codebook (stage 1) 1st subframe	3
GB1	Gain codebook (stage 2) 1st subframe	4
P2	Pitch delay 2nd subframe	5
S2	Signs of fixed-codebook pulses 2nd subframe	4
C2	Fixed codebook 2nd subframe	13
GA2	Gain codebook (stage 1) 2nd subframe	3
GB2	Gain codebook (stage 2) 2nd subframe	4

From the proposed model, it is suitable for using in low-power, portable devices, enabling longer battery life. This is because the computational load can be reduced by encrypting for only 45% of the whole bit streams. Moreover, the security of voice data is equivalent to full encryption [1].

2.2.2 HDSP (Hierarchical Data Security Protection)

HDSP scheme [3] is an approach to prevent voice conversation from eavesdropping, especially for conversation over the VoIP system. And an operation of this is the following. At first, encoded inter-frames of voice data are interleaved to solve the problem of continuous packet loss. Then, the interleaved voice data are encrypted. However, in the case of the inter-frame data interleaving, frame re-ordering technique based on a chaotic bit-string is employed. As for intra-frame data encryption, pixel value transformation technique is used in bit swapping and then XOR operation is used, based on the chaotic bit-string. And the details of their operations are illustrated in Figure 2.3.

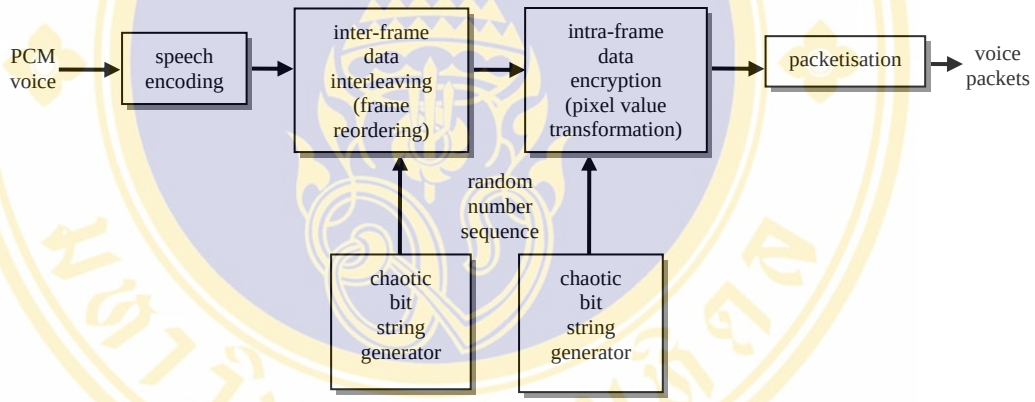

Figure 2.3 Operations of inter-frame data interleaving and intra-frame data encryption in HDSP scheme

As for the above model [3], it can be expanded to make more understanding as follows. As for the inter-frame data interleaving, it can prevent the continuous packet loss. For example, if the frame sequence in each packet: 1, 2, 3, 4, and 5, is sent over the network and lost, data reconstruction is difficult in such a case. On the other hand, if the data interleaving is performed before sending the data over the network, for example, by swapping the frame sequence of the data to 3, 9, 10, 7, and 1, the receiver will recognize that the lost frames are not continuous in natural order in

case of data loss. Error recovery technique is then employed to reconstruct the voice data, providing better quality of voice signals.

Another point that should be addressed here is the intra-frame data encryption by means of bit swapping inside a pair of data, that is, (0, 4), (1, 5), (2, 6), and (3, 7), generated by a chaotic bit-string generator. And then, these four chaotic bits from the bit-string generator are XORed with the 1st, 3rd, 5th, and 7th data from the bit swapping. From this scheme, it can help preventing voice eavesdropping during the conversation.

2.2.3 SRTP (Secure Real-time Transport Protocol)

SRTP [4] is the standard used in eavesdropping protection in RTP (Real-time Transport Protocol) traffic. It is a security protocol developed by IETF (Internet Engineering Task Force). It is suitable for preventing only voices, or both images and voices transmitted over RTP protocol from eavesdropping. Also, this can be applied to real-time applications (streaming and conversational multimedia) [5]. As mentioned earlier, other than the confidentiality of RTP payload, the integrity of RTP packet is also taken into consideration as well.

In terms of an operation of SRTP, it can be explained as follows. SRTP encrypts RTP payload data with AES (Advanced Encryption Standard) algorithm as shown in SRTP packet in Figure 2.4 below. Nevertheless, before encryption, the key has to be exchanged first, in other words, key management by means of MIKEY (Multimedia Internet KEYing) algorithm has been performed.

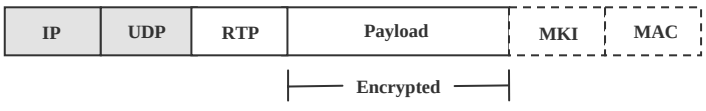

Figure 2.4 Overview of an SRTP packet

According to the above-mentioned protocol, it is properly used in preventing both wired and wireless communications from eavesdropping. This is because the design for this protocol is aimed at not only providing the security, but also using in

the systems with low-computational cost, low-bandwidth cost, independence from the underlying transport, network, and physical layers used by RTP [4].

2.2.4 Security for GSM network

In this section, the eavesdropping protection of the Global System for Mobile communications (GSM) network [6] [7] is presented as shown in Figure 2.5: Encryption of GSM.

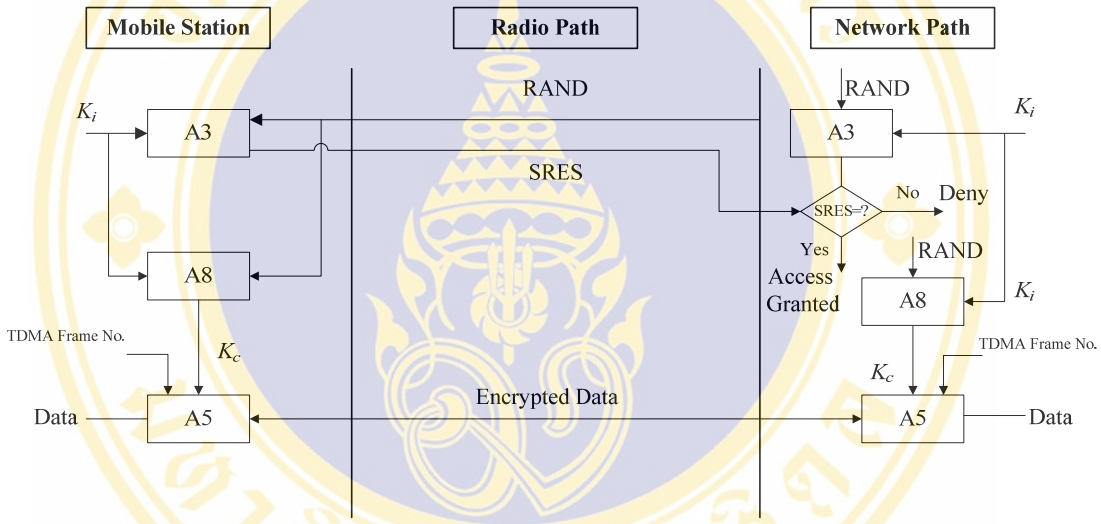

Figure 2.5 Encryption of GSM

As illustrated in Figure 2.5, the security system starts with an authentication procedure. When the communication is linked between MS (Mobile Station) and the network, the network will send RAND (random number) to the MS. After that, the MS uses K_i (individual subscriber authentication key) stored in the subscriber ID (SIM) in A3 (authentication algorithm) to produce SRES (signed response) and send it back to the network. And this SRES is compared with the SRES generated by the network itself. At the same time, the MS derives key K_c (ciphering key) from RAND to use it in the encryption and decryption processes further. Also, the network will use K_c in the encryption and decryption algorithms.

From the mentioned security system, the voice data are encrypted between the MS and the network only. In this case, the GSM system is not an end-to-end confidentiality service [7]. Moreover, the key used in the encryption process is determined by the network operator.

2.2.5 End-to-end security for GSM users

As the GSM system encrypts the data only on the air link between the MS and BTS (Base Transceiver Station). As for the link between the BTS and MSC (Mobile Switching Center), it is unencrypted that can be easily eavesdropped. So, this research proposes the end-to-end security for GSM users [8]. In such a case, the mobile phones need GSM_CRYPT module to encode and encrypt the voice data. The reason why we need this module is that the GSM system has already had the standard protocols for digitization of voice, compression of voice data and air link encryption. So, the voice channel cannot be used in the end-to-end security. Therefore, in this research [8], the data channel of GSM is used to route the voice as data as illustrated in Figure 2.6.

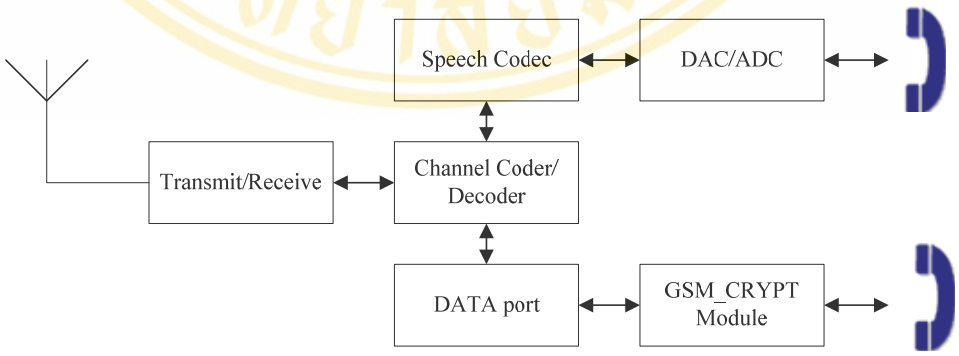

Figure 2.6 Secure GSM mobile unit

2.3 Objective

The objective of this research is to develop an approach to prevent voice conversation over the network from eavesdropping, for example, the conversation over the mobile phone networks as well as the Internet networks like VoIP. By this way, the developed system must be the end-to-end security that users control the encryption keys by themselves, creating the real-privacy system.

2.4 Problem Statements

In voice communications over the network system such as analog and digital, voice data signals can be easily tapped to use in eavesdropping, for example, communications over the mobile phones or the Internet-based communications. The voice data transmitted over digital systems are in the form of compressed or encoded voice data and can be tapped, decoded and played back, resulting in the intercepted data. So, the problem of this research is to find the effective approach to protect the encoded voice data against eavesdropping before transmitting over the network.

2.5 Problem Scope

In corresponding to the current problems and to define the clear topics of the study, in this section, we propose the scope of this research, covering the following items:

1. Develop the approach to protect voice conversation over digital networks against eavesdropping by, however, covering only the encoded voice data using the G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards.
2. This research is also limited only to point-to-point communications.

CHAPTER III
CONCEPTUAL DESIGN

From problem statements described in Chapter II, this chapter presents the conceptual design used in preventing voice conversation from being tapped over the network such as the mobile phone systems, or the Internet-based communications.

3.1 Conceptual Design of Eavesdropping Protection

3.1.1 Concept of voice data encryption

In conversations in the network systems, the voice data sent over the network systems are in the form of encoded voice data e.g. using the G.729 [2] and G.723.1 [9] encoding standards and so on. So, the eavesdroppers can tap the voice data signals and decode the voice data with the mentioned encoding standards and then play them back, resulting in eavesdropping finally.

Therefore, this research is aimed at solving this problem by encrypting the encoded voice data before sending over the network and then decrypting at the terminal. In doing so, the encryption and decryption keys are specified by the two users, not by the network operators. This is to prevent the encoded voice data transmitted over the network from eavesdropping. Although the voice data can be tapped over the network and then decoded, they are not easily understood. Moreover, this scheme takes less encryption time since the encoded voice data are lessened.

However, the encoded voice data before sending over the network are in the form of packets. Hence, in this work, partial- frame encryption scheme is investigated to be another choice in preventing eavesdropping, assuming the partially-encrypted voice data to be packet error. So, when the data are decoded, they cannot be understood as in the case of all frame encryption. Beyond that, partial frame encryption takes less time in encryption and decryption than all frame encryption.

Furthermore, this partial frame encryption can help increasing the degree of difficulty in discovering the tapped voice data or key, or so-called cryptanalysis.

3.1.2 Concept of key management

In this work, a symmetric encryption is presented, sending the keys between the parties in a different signal channel from the signal channel of the conversation or using other media such as E-mail, SMS, etc. This is to avoid the keys from being used in eavesdropping. From this, only the communicating parties know the keys that create the real-privacy circumstance.

At present, the mobile phone system uses the key in voice data encryption and then sends the data over the network system. And the system uses the same key as the key of the speaker in decryption and sends the data over the network that is not the real-privacy situation. To solve this problem, the proposed scheme can make the personalized security on the mobile phones and can be implemented without having any effects on the conventional network systems.

As above-discussed conceptual design, we present the model of voice data transformation for preventing the eavesdropping on the conversation as carefully described in the following model.

3.2 Voice Data Transformation Model

To achieve the set goal, we propose the voice data transformation model as illustrated in Figure 3.1.

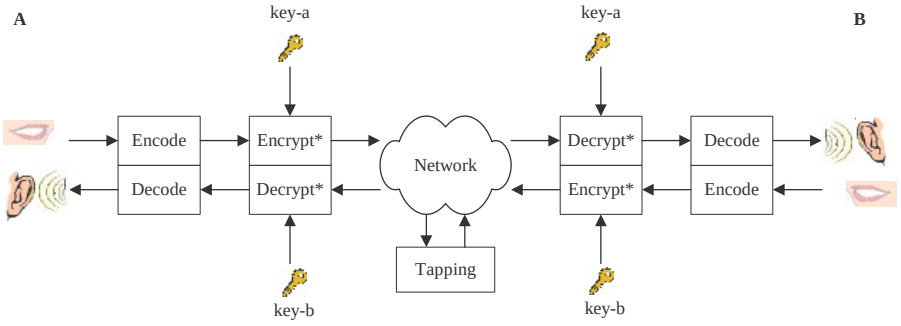

Figure 3.1 Voice data transformation model

From Figure 3.1, we can describe how the voice data transformation model resolves the specified problems as follows. The process begins with encoding the A/D (Analog-to-Digital) voice data. Then, the encoded voice data are partitioned into blocks and encrypted with the key (Encrypt* means 1-to-M block encryption in this model).

Referring to the above model, the voice data are transformed into the uneavesdropped data and then sent over the network to the terminal receiver. After that, the receiver decrypts the voice data, using the key as well (Decrypt* means 1-to-M block decryption in this model). And finally, the voice data are decoded before transforming back to the voice signals that can be understood as the original voices. However, in real practice, the eavesdropping protection consists of two processes: key exchange setup as well as encryption and decryption procedures. And the details of them are described in greater detail in sections 3.2.1 and 3.2.2, respectively.

3.2.1 Key exchange setup

In the following, we present the process of key exchange between the communicating parties: user A and user B. By this way, we use a sending key of 64 bits in the encryption and decryption schemes according to DES (Data Encryption Standard) algorithm.

A key exchange setup is a key delivery between user A and user B via SMS (Short Message System) as shown in Figure 3.2. In this case, the key is sent via the different signal channel from the signal channel of the conversation to avoid the key from being used in eavesdropping.

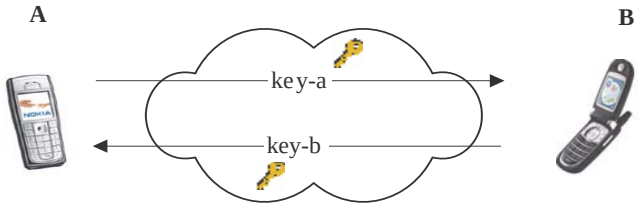

Figure 3.2 Delivering a key between user A and user B

From Figure 3.2, the key exchange setup begins with user A's sending device sends key-a to user B's receiving device. Next, user B's receiving device records the received key in the phonebook that is user A's (the transmitter's). In reverse order, user B's sending device sends key-b to user A's receiving device. At the receiver side, user A's receiving device also records the received key (key-b) in the phonebook that is user B's (the transmitter's). However, in practice, there are also phone calls from other phones. In such a case, the receiving devices also exchange the keys with those phones in advance and record their keys as well as storing in the phonebook of each individual person. Thus, in case there is any phone call coming in, the receiving devices will search for the key used in decryption in association with the coming call, based on an existing phonebook.

As mentioned earlier, the proposed key exchange setup is suitable for using in communications system that does not need key distribution centers in key exchange. For example, we can adopt this to the mobile phone systems; their network systems have no need to perform the key exchange as mentioned.

3.2.2 Encryption and decryption

In this section, we introduce the encryption and decryption procedures of voice data with a symmetric key. That is to say, the sender needs the same key used in the encryption process as the key used in the decryption process of the terminal receiver. And to enable standardization for widespread use, an algorithm exposure is recommended such as the DES algorithm, configured for a 64-bit key as a secret key.

As above-mentioned, the encryption and decryption start with user A's sending device partitions the packet encoded voice data into blocks, 64 bits each. Then, each block of the data is encrypted with 1-to-M block (M is the number of blocks in each group of the packets). That is to say, the first block of each group is encrypted with the key while the next block(s) (M-1 block) is (are) not encrypted as shown in Figure 3.3: examples of 1-to-2 blocks and 1-to-3 blocks encryptions (the blocks shown in black color are encrypted blocks) in order. In this case, we use the DES algorithm in encryption by using key-a derived from the key exchange setup. And after the voice data are transformed and sent to user B's receiving device, at the

receiver's side, user B's receiving device decrypts the data with key-a and transforms them into the encoded voice data as shown in Figure 3.4.

On the contrary, when user B's device acts as the sender, user B's sending device does the same way as user A's receiving device. However, user B's sending device uses key-b in encryption while user A's receiving device uses key-b in decryption.

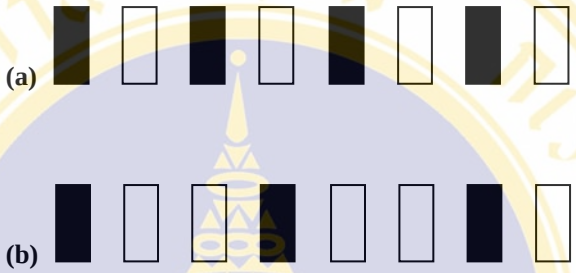

Figure 3.3 Encrypted encoded voice data of 1-to-2 blocks is shown in (a) and 1-to-3 blocks is illustrated in (b).

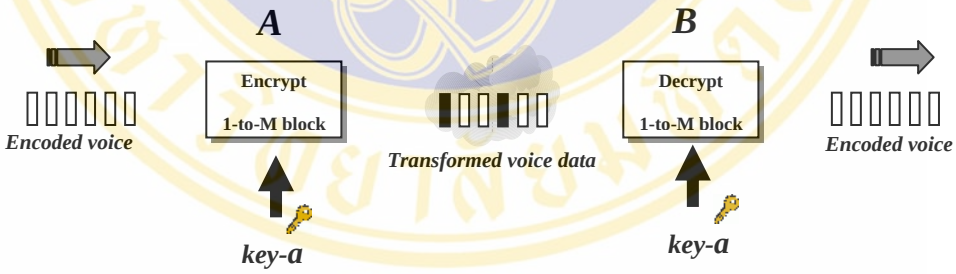

Figure 3.4 Encoded voice data encryption

In practice, the communicating parties do not need to use the same number of the encrypted and unencrypted blocks. In other words, each party can choose a different ratio. And in the real-conversation scenarios, they can specify the ratio of the encrypted and unencrypted blocks for 1-to-M block by themselves such as 1:1, 1:2, 1:3, 1:4, or 1:5. And a calculation to find M value is shown in Section 3.3. From this, the packet voice data sent over the network are encrypted. So, with this proposed scheme, the eavesdroppers cannot know which ones are encrypted and which ones are

unencrypted that is difficult for the cryptanalysis. In addition, by this technique, the communicating parties keep the ratio of 1-to-M block in secret, by making a common agreement in the process of call-setup before encryption.

3.3 Calculation of M Value

According to the proposed principle, when there is the packet error happening with the encoded voice data sent over the network, the quality of voice at the receiving terminal is lower and it cannot be understood. So, we apply this to the eavesdropping protection by comparing the encrypted encoded voice data with the packet error or the packet loss. Then, the calculation of the smallest number to be encrypted is presented in order to find the most suitable value that still preserves the voice data security. In this case, we use a formula for calculating: the E-model from G.107 [10] to find the quality of voice or a MOS (Mean Opinion Score) derived from a transmission rating factor R as shown in the following equations:

$$\begin{aligned} &\textbf{For } R<0: && MOS = 1 \\ &\textbf{For } 0<R<100 && MOS = 1+0.035R+R(R-60)(100-R)7*10^{-6} \\ &\textbf{For } R>100 && MOS = 4.5 \end{aligned} \quad (1)$$

Here, R value is related to the packet loss as shown in the following equations.

$$R = R_o - I_s - I_d - I_e + A \quad (2)$$

where R_o is Basic signal-to-noise ratio
I_s is Simultaneous impairment factor
I_d is Delay impairment factor
I_e is Effective equipment impairment factor
A is Advantage factor

$$I_e = I_m + ((95-I_m) * P_e / ((P_e/ B) + P_r)) \quad (3)$$

where I_m is Equipment impairment factor
P_e is Packet-loss probability
P_r is Packet-loss robustness factor
B is Burst ratio

From the above equations, we can determine the percentage of the packet loss of G.729 encoded voice that provides low voice quality and unknown voice. From this, MOS of the voice after transformation is less than 2 that is not understood. And in the model, we place MOS=1.6 in an equation 1, obtaining R=29.82. And then, place R value in an equation 2, together with placing the sum of R_o and I_s variables (2 terms: intrinsic quality) that is 84.3 [11] in the equation 2. As for I_d and A variables, they are aside from the condition, so they are not used in this calculation. After that, we obtain I_e=54.47 and then place this value in an equation 3, together with I_m=10, B=1, and P_r=19 [12] [13]. Finally, the packet loss from the calculation is about 20%. And from the calculated packet loss we find that at least 20% of the encrypted voice data cannot be intercepted.

Hence, we apply this principle to the 1-to-M block encryption; M is an integer number from 1 to 5. Otherwise, encrypt 1 block from 5 blocks, amounting to 20%. Also, this technique is used for G.723.1, G.726 [14] and AMR [15] encoded voice data.

CHAPTER IV
SYSTEM DESIGN

4.1 System Design

According to our proposed model in the previous chapter, the following is the detailed design of eavesdropping protection. An overview of the system design is presented by the structure chart in Figure 4.1. And from this figure, the voice data transformation system consists of two main processes: Transmit and Receive processes. According to these processes, Transmit process transforms the voice data before transmitting over the network system, while Receive process is in the listener's receiving section, receiving the transformed voice data from the network system and converting them back into the original speakers' voice data. And these processes are corresponding to the sequence diagrams of the system as shown in Figures 4.2 and 4.3.

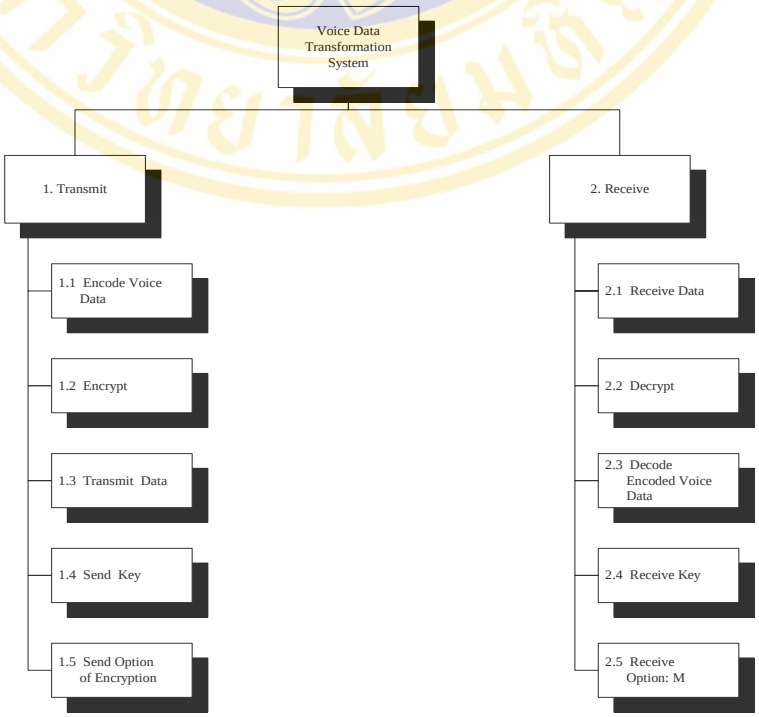

Figure 4.1 Structure chart of the voice data transformation system

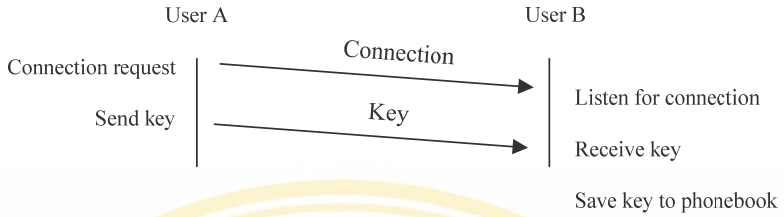

Figure 4.2 Key exchange process

From Figure 4.2, the process of key exchange is indicated, starting with exchanging the key by the communicating parties before sending voice data over the network. At this point, the speaker sends the key used in encryption to the listener. And the key is used in decrypting the transformed voice data further. As shown in Figure 4.3, the voice data are transmitted over the voice data transformation system, while user A establishes the connection and then encryption. And next, user A sends the option of encryption, the encrypted M value, to user B. On the other hand, user B sends his/her determined encrypted M value to user A. Then, the voice data are encoded and the encoded voice data are encrypted. After that, the data are sent over the network to the other party. And when each party receives the packets of data, they will decrypt and decode them to obtain the speaker's voice data as well as transforming the voice data into the voice signals.

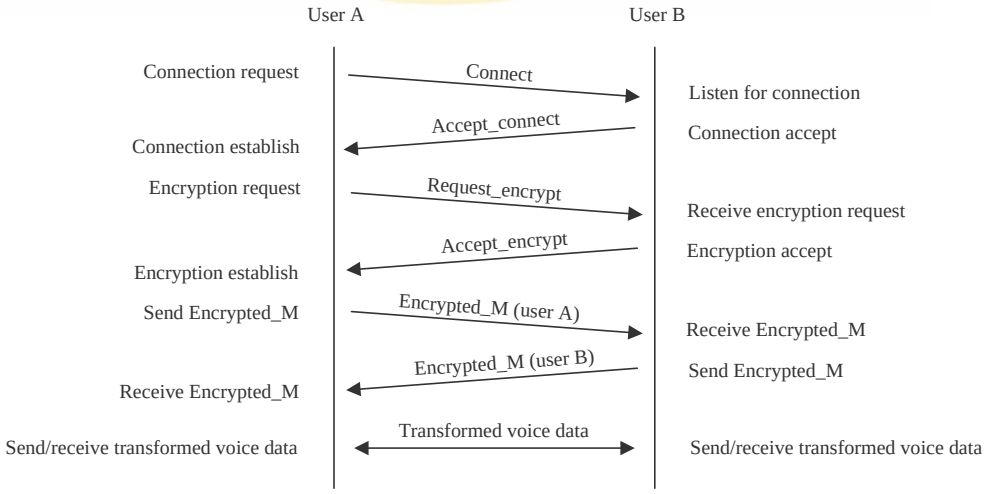

Figure 4.3 Voice data transformation system process

4.1.1 Subsystem processes

1. Transmit

This process transforms the voice data of the speaker into the uneavesdropped voice data before sending over the network system. By this way, the process consists of five following subsystems: Encode Voice Data, Encrypt, Transmit Data, Send Key, and Send Option of Encryption.

1.1 Encode Voice Data

Definition: A process that is used to receive the voice data from the A/D conversion to encode with the G.729, G.723.1, G.726 or AMR standards. From this, the amount of voice data is reduced. After that, the encoded voice data received from this process are transmitted to the process of encryption as shown in Figure 4.4, Encode Voice Data flowchart.

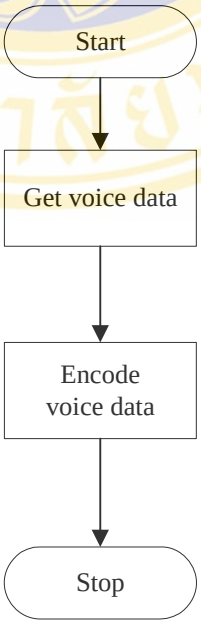

Figure 4.4 Encode Voice Data flowchart

Pseudo code:

Get VOICE_DATA;

Encode VOICE_DATA;

1.2 Encrypt

Definition: A process that is used to encrypt the voice data derived from the Encode Voice Data process. In doing so, the encoded voice data are divided into blocks, 64 bits each. Then, the data are encrypted using the DES algorithm with a 64-bit key from the phonebook. In the encryption process, M value from the selection of the speaker is used in the 1-to-M block encryption, ranging from 1 to 5. In other words, the first block of each packet is encrypted, while M-1 block is unencrypted, alternating like this until the end of the conversation. At the same time, the blocks of the transformed voice data are stored in the buffer before sending over the network as illustrated in Figure 4.5, Encrypt flowchart.

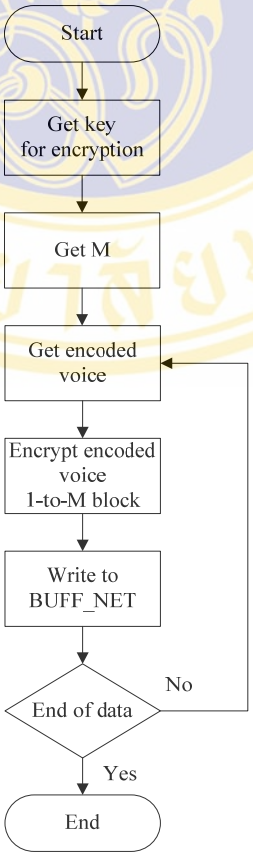

Figure 4.5 Encrypt flowchart

Pseudo code:

```
Get KEY from PHONEBOOK by matching to the communicating
parties;
Get M;
I=1;
J=0;
Get ENCODED_VOICE_DATA to BUFF;
Do until the end of data
    If   I>J Then (
    Encrypt [ENCODED_VOICE_DATA (BUFF)] by KEY;
    J=J+M;
    );
    Write to BUFF_NET;
    I + +;
ENDDO
```

1.3 Transmit Data

Definition: A process that is used to transmit the transformed voice data, being stored in the buffer in the Encrypt process, over the network system.

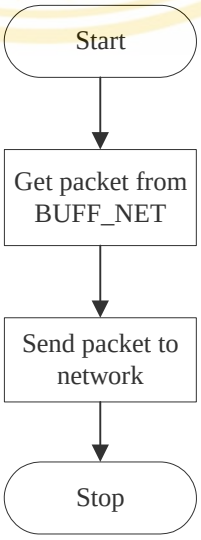

Figure 4.6 Transmit Data flowchart

Pseudo code:

Get TRANSFORMED_VOICE_DATA from BUFF_NET;

Send PACKET_TRANSFORMED_VOICE_DATA to NETWORK;

1.4 Send Key

Definition: A process that is used to generate the encryption key and save the key in the phonebook as well as sending the key to the communicating party to use in the decryption process further.

Figure 4.7 Send Key flowchart

Pseudo code:

Get SEND_KEY_COMMAND from USER;

Generate KEY by Random KEY;

Save KEY to PHONEBOOK;

Send KEY to Receiver;

1.5 Send Option of Encryption

Definition: A process that is used to receive the option of encryption, M value, to use in the 1-to-M block encryption in the Encrypt process. Besides, M value is encrypted with the key and sent to the communicating party to use in the decryption process further.

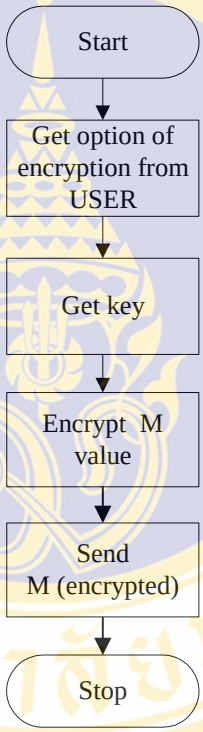

Figure 4.8 Send Option of Encryption flowchart

Pseudo code:

Get M from USER;

Get KEY;

Encrypt M by KEY;

Send ENCRYPTED_M to Receiver;

2. Receive

This process shows an operation of the receiving section by converting the transformed voice data back into the original voice data before being encoded in the transmission section. The process is composed of five subsystems: Decrypt, Decode Encoded Voice Data, Receive Data, Receive Key, and Receive Option: M. And the following is the details of each process.

2.1 Receive Data

Definition: A process that is used to store the transformed voice data, from the received packet over the network, in the buffer to further decrypt in the process of Decrypt.

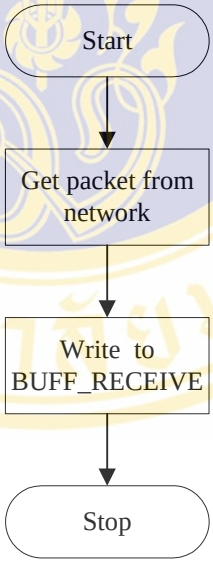

Figure 4.9 Receive Data flowchart

Pseudo code:

Get PACKET_NETWORK;
Get TRANSFORMED_VOICE_DATA from PACKET_NETWORK;
Write TRANSFORMED_VOICE_DATA to BUFF_RECEIVE;

2.2 Decrypt

Definition: A process that is used to decrypt the transformed voice data from the buffer by using the key from the phonebook. And this key is received from the speaker in the key exchange setup. Then, M value from the speaker is used in the 1-to-M block decryption the same as in the Encrypt process. Afterwards, the blocks of the voice data are stored in the buffer to further use in the Decode Encoded Voice Data process.

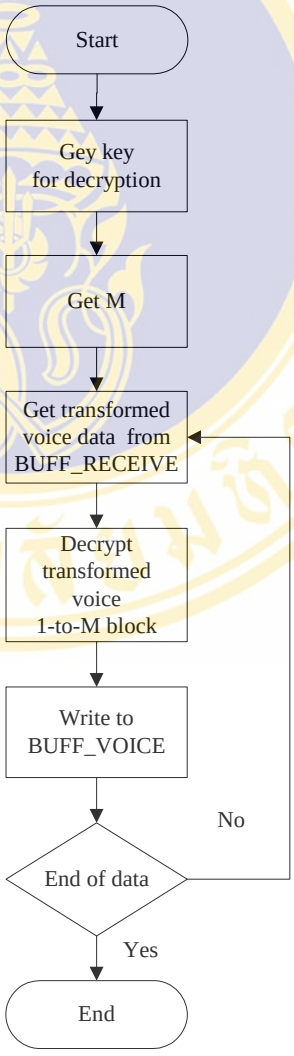

Figure 4.10 Decrypt flowchart

Pseudo code:

Get KEY from PHONEBOOK by matching to the communicating parties;
Get M;
I=1;
J=0;
Get TRANSFORMED_VOICE_DATA from BUFF_RECEIVE to BUFF;
Do until the end of data
If I > J Then (
Decrypt [TRANSFORMED_VOICE_DATA (BUFF)] by KEY;
J=J+M;
);
I + +;
Write to BUFF_VOICE;
ENDDO

2.3 Decode Encoded Voice Data

Definition: A process that is used to decode the encoded voice data by using the same standard as in encoding, consequently receiving the same voice data.

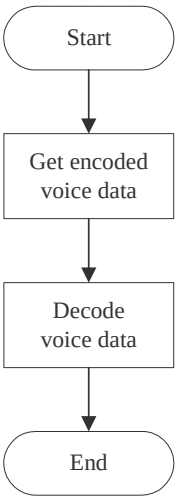

Figure 4.11 Decode Encoded Voice Data flowchart

Pseudo code:

Get ENCODED_VOICE_DATA from BUFF_VOICE;

Decode ENCODED_VOICE_DATA;

2.4 Receive Key

Definition: A process that is used to receive the key from the speaker in the key exchange setup and then save in the phonebook to use in decrypting the voice data of the speaker.

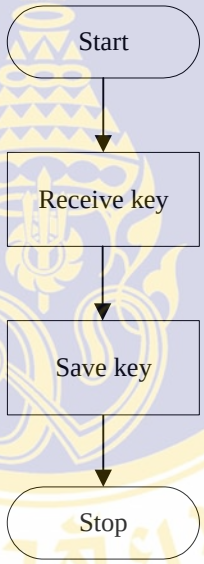

Figure 4.12 Receive Key flowchart

Pseudo code:

Receive KEY;

Save KEY to PHONEBOOK;

2.5 Receive Option: M

Definition: A process that is used to receive the option of encryption that is the encrypted M value of the speaker in the Send Option of Encryption process. Then, M

value is decrypted with the key and M value is used to decrypt the transformed voice data in the 1-to-M block decryption further.

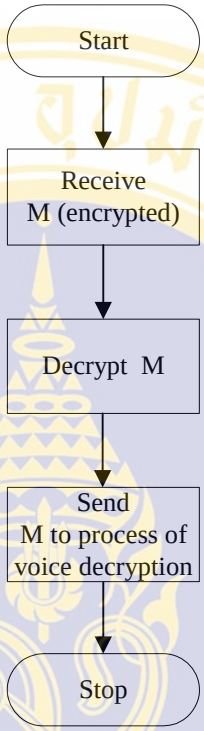

Figure 4.13 Receive Option: M flowchart

Pseudo code:

Receive ENCRYPTED_M;

Get KEY;

Decrypt ENCRYPTED_M by KEY;

Send M to Decrypt Process;

CHAPTER V
SYSTEM IMPLEMENTATION

5.1 Introduction

In this section, the implementation of the system prototype of voice data transformation is described. As mentioned earlier, the system implementation is divided into two main sections: key exchange and communication as illustrated in Figures 5.1 and 5.2, respectively. By this way, we have implemented a one-way conversation, that is, user A is the speaker and user B is the listener.

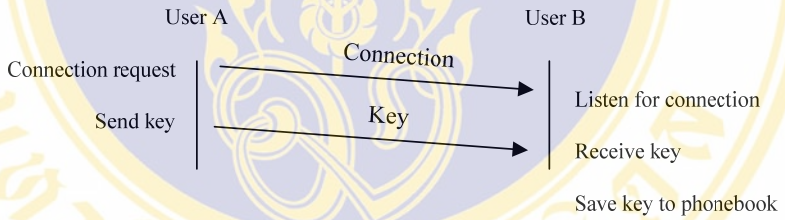

Figure 5.1 Key exchange testing process

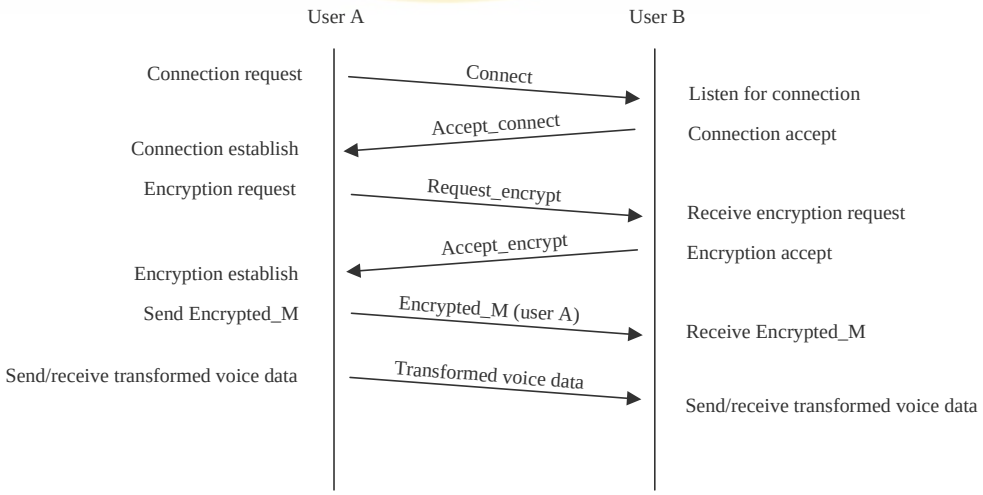

Figure 5.2 Voice data transformation system testing process

From Figure 5.1, the implementation of key exchange is indicated, starting with exchanging the key by the communicating parties before sending over the voice data transformation system. And the key is used in decrypting the transformed voice data further. At this point, the speaker, user A, sends the key used in encryption to the listener, that is, user B, over TCP/IP network. And when user B receives the key, user B then saves the key.

From Figure 5.2, the voice data are transmitted over the voice data transformation system, while user A establishes the connection and then encryption. And next, user A sends the option of encryption, the encrypted M value, to user B. Then, the voice data are encoded and the encoded voice data are encrypted. After that, the data are sent over the network to user B. And when user B receives the packets of data sent from user A, user B will decrypt and decode them to obtain the voice data of user A as well as transforming the voice data into the voice signals.

In addition, in this process, the implementation of eavesdropping simulation is employed by using the transformed voice data received by user B to decode immediately without decrypting. And then, the voice data received from the eavesdropping simulation is listened.

5.2 Program Module

In program coding, we develop the programs using C language. Each module of the programs is corresponding to subsystem processes as shown in Appendix. At the same time, the program coding is also illustrated in Appendix.

5.3 Platform

We have implemented the voice data transformation system on computer hardwares and softwares as follows.

1. Computer Hardwares: Notebook Computer, CPU Intel, Pentium III-650 MHz, memory 128 MB
2. Computer Softwares:

- Operating System : Windows Me
- Compiler : Ms Visual C
- API modules:
  - G.729, AMR : download from www.voiceage.com
  - G.723.1, G.726 : download from www.intel.com
  - DES : download from www.slavasoft.com

5.4 System Testing

5.4.1 Simulation

In implementing the system, we simulate a PC (Personal Computer) as a sending device of the speaker and a receiving device of the listener simultaneously. Also, we simulate the TCP/IP network system as the mobile phone network or the Internet network. In this manner, the data are sent over the TCP/IP network. However, we need to communicate over a loop-back IP address in case of sending-receiving devices as shown in Figure 5.3. There is a process of receiving the data sent from the network. This process listens for the connection on TCP Port. Also, the process acting as the sending device to send the data to the TCP Port of the receiving device is included. This is according to the implementing process as shown in Section 5.1.

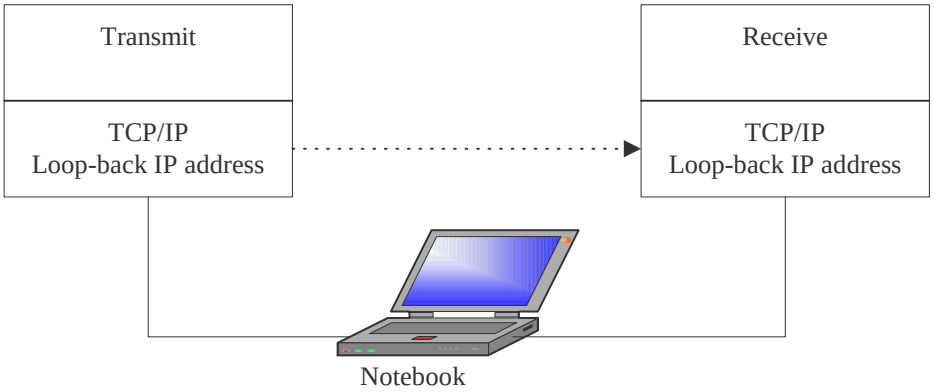

Figure 5.3 Simulation of voice data transformation system

CHAPTER VI
EXPERIMENTAL RESULTS

In this experiment, we have studied the approach for voice eavesdropping protection. In other words, this research investigates the encryption of voice data transmitted over the network such as the Internet, VoIP, or the mobile phone system. This is to analyze that which is the best solution to transform voice data: *all* frame (encryption for all blocks of voice data) or *partial* frame (1-to-M block encryption; M is the number of blocks encrypted at the first block) encryption. In this regard, the time used in speech encryption and decryption has been taken into consideration. And the one that takes less time to do so means that it enables longer battery life, especially in mobile, portable devices such as the mobile phones that need limited battery power consumption. At the same time, we also consider the security from eavesdropping after encryption.

As above-mentioned, this experiment is organized into three main sections as follows. Section I provides setting up key exchange to use in speech encryption and decryption. Section II presents a variety of encryption schemes for G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] encoded voice data. This is to analyze that which encryption scheme takes less time and preserves speech security. And in Section III, the appropriate encryption scheme derived from Section II is tested together with key exchange by simulating the eavesdropping circumstance.

6.1 Key Exchange Setup

In the following, the test of key exchange between the communicating parties: user A and user B, is presented. The sending key of 64 bits, for example, key "&%TyS#hM", is used in encryption and decryption according to the DES algorithm.

Then, this approach is further used together with the finding of the experiment in Section II.

In this scenario, we use one computer in the simulation, acting as two mobile phones: the sending and the receiving devices. The key is then sent from the sending device via SMS to the receiving device as shown in Figures 6.1 and 6.2 below. In this case, the key is sent via the different signal channel from the signal channel of the conversation to avoid the key from being used in eavesdropping.

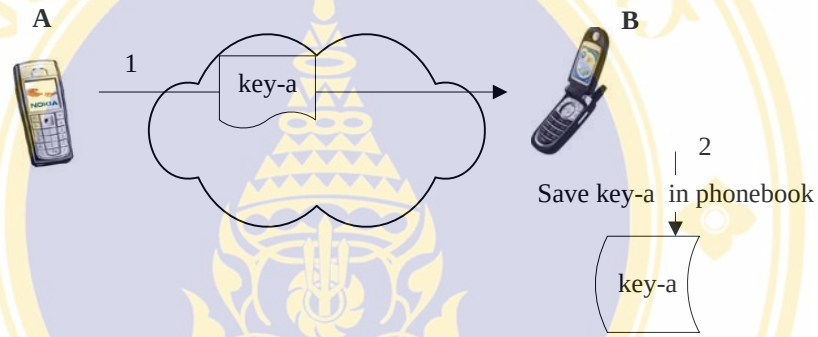

Figure 6.1 Sending key in encryption from A to B

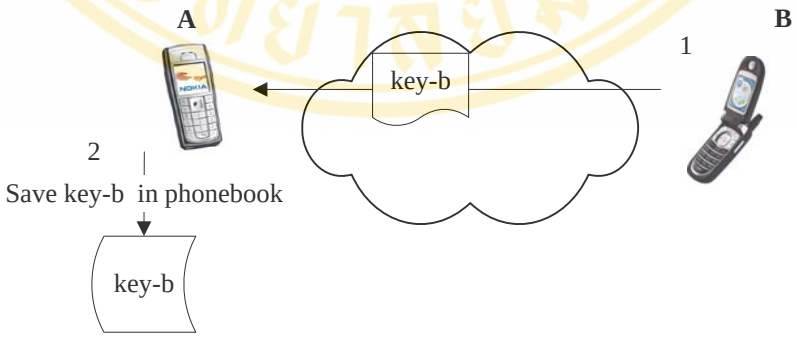

Figure 6.2 Sending key in encryption from B to A

From above two simulated figures, the experiment begins with the sending device (by user A) sends key-a to user B's receiving device. Next, user B's receiving device records the received key in the phonebook that is user A's (the sender's). In the same manner, user B's sending device sends key-b to user A's receiving device. At the receiver's side, user A's receiving device also records the received key (key-b) in the phonebook that is user B's (the sender's). However, in practice or in fact, there are also phone calls from other phones. In such a case, the receiving devices also exchange the keys with those phones in advance. And then, the keys are recorded as well as stored in the phonebook of each individual person. Thus, in case there is any phone call coming in, the receiving devices will search for the key used in decryption in association with the coming call, based on the existing phonebook.

6.2 Encryptions of G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] Encoded Voice Data

In the following section, the various forms of encryption scheme are used with G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] encoded voice data. This is to analyze that which speech encryption takes less time and also achieves voice data security. And the details are as follows.

6.2.1 Encryption of G.729 encoded voice data

Here, a variety of encryptions for G.729 encoded voice data are tested as illustrated in the following three processes below.

6.2.1.1 Preparatory process for G.729 speech encoding standard

At the beginning, speech is recorded for five minutes in the form of raw 16-bit mono PCM at a random frequency of 8,000 Hz. And after recording, the file size is 4,800,000 bytes. Then, this file is encoded using speech encoding algorithm, the ITU-T G.729 standard at 8 kb/s, producing the new file size of 300,000 bytes to use in the next experimental process.

6.2.1.2 All frame encryption

There are three processes in speech encryption and decryption as shown in the following.

Step 1

At first, encrypt the file from the above preparatory process with the DES algorithm block by block (1 block has 8 bytes or 64 bits) until all blocks in the file have been completely encrypted. Take the time from the beginning to the end of encrypting all the blocks in the file. Then, record the timing result in Table 6.1.

Step 2

This step is the simulated eavesdropping. That is to say, the encrypted file from step 1 is decoded and then we listen to this decoded file by using Sound Forge 4.5h software to open the file. Listen to the speech and record the listening result in Table 6.2.

Step 3

This step is the process of normal decryption and decoding by using the same file from step 1 to decrypt. At the same time, take the whole time of decrypting and record the timing result in Table 6.1. Then, decode the file derived from the decryption process and open it by using the Sound Forge 4.5h software. At last, listen to the speech and record the listening result in Table 6.3.

6.2.1.3 1-to-M block encryption (M = 2, 3, 6, 11 in order)

In this section, we perform experiments according to all three processes below. However, in each round of the experimental processes, let M be the block number, we substitute M value for 2, 3, 6, and 11, respectively. And the details are shown in the following.

Step 1

In the beginning, encrypt the file from the preparatory process with the 1-to-M block encryption. Take the time from the beginning to the end of this encryption. Then, record the timing result in Table 6.1.

Step 2

This step is the simulated eavesdropping. That is to say, the encrypted file from step 1 is decoded and then we listen to this decoded file by using the Sound Forge 4.5h software to open the file. Listen to the speech and record the listening result in Table 6.2.

Step 3

This step is the process of normal decryption and decoding by using the same file from step 1 to decrypt. At the same time, take the whole time of decrypting and record the timing result in Table 6.1. Then, decode the file derived from the decryption process and open it by using the Sound Forge 4.5h software. And finally, listen to the speech and record the listening result in Table 6.3.

Table 6.1 Encryption and decryption times of G.729 speech encoding standard

Encrypt / Decrypt	Encryption Time (seconds)	Decryption Time (seconds)
All blocks	0.11	0.10
1 block-to-2 blocks	0.06	0.05
1 block-to-3 blocks	0.05	0.05
1 block-to-6 blocks	0.02	0.02
1 block-to-11 blocks	0.02	0.02

Table 6.2 The listening results of encryptions for G.729 speech encoding standard in step 2

Encrypt	Listening Results
All blocks	1
1 block-to-2 blocks	1
1 block-to-3 blocks	1
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

Table 6.3 The listening results of encryptions for G.729 speech encoding standard in step 3

Encrypt	Listening Results
All blocks	4
1 block-to-2 blocks	4
1 block-to-3 blocks	4
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

6.2.2 Encryption of G.723.1 [6.3 kb/s] encoded voice data

In this experiment, we study the change of encryption period of G.723.1 [6.3 kb/s] encoded voice data and then decode the file. After that, listen to the speech using the Sound Forge 4.5h software. By the way, the experimental processes are the same as in section 6.2.1

Nevertheless, there is a difference in terms of the preparatory process. That is, encoding the recorded file of 4,800,000 bytes with speech encoding algorithm, the ITU-T G.723.1 standard at 6.3 kb/s and saving the new file size of 240,000 bytes to use in the next experimental process. The details of the experimental results are shown in Tables 6.4, 6.5, and 6.6.

Table 6.4 Encryption and decryption times of G.723.1 [6.3 kb/s] speech encoding standard

Encrypt / Decrypt	Encryption Time (seconds)	Decryption Time (seconds)
All blocks	0.08	0.08
1 block-to-2 blocks	0.05	0.04
1 block-to-3 blocks	0.03	0.03
1 block-to-6 blocks	0.02	0.02
1 block-to-11 blocks	0.02	0.02

Table 6.5 The listening results of encryptions for G.723.1 [6.3 kb/s] speech encoding standard in step 2

Encrypt	Listening Results
All blocks	1
1 block-to-2 blocks	1
1 block-to-3 blocks	1
1 block-to-6 blocks	2
1 block-to-11 blocks	3

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

Table 6.6 The listening results of encryptions for G.723.1 [6.3 kb/s] speech encoding standard in step 3

Encrypt	Listening Results
All blocks	4
1 block-to-2 blocks	4
1 block-to-3 blocks	4
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

6.2.3 Encryption of G.723.1 [5.3 kb/s] encoded voice data

As well as the experiments in section 6.2.2, in the following, we study the change of encryption time of G.723.1 [5.3 kb/s] encoded voice data and then decode the file. After that, listen to the speech using the Sound Forge 4.5h software. By the way, the experimental processes are the same as in section 6.2.1

However, there is a difference in terms of the preparatory process. That is, encoding the recorded file of 4,800,000 bytes with speech encoding algorithm, the ITU-T G.723.1 standard at 5.3 kb/s and saving the new file size of 200,000 bytes to use in the next experimental process. The details of the experimental results are shown in Tables 6.7, 6.8, and 6.9.

Table 6.7 Encryption and decryption times of G.723.1 [5.3 kb/s] speech encoding standard

Encrypt / Decrypt	Encryption Time (seconds)	Decryption Time (seconds)
All blocks	0.07	0.07
1 block-to-2 blocks	0.04	0.04
1 block-to-3 blocks	0.03	0.03
1 block-to-6 blocks	0.02	0.02
1 block-to-11 blocks	0.01	0.01

Table 6.8 The listening results of encryptions for G.723.1 [5.3 kb/s] speech encoding standard in step 2

Encrypt	Listening Results
All blocks	1
1 block-to-2 blocks	1
1 block-to-3 blocks	1
1 block-to-6 blocks	2
1 block-to-11 blocks	3

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

Table 6.9 The listening results of encryptions for G.723.1 [5.3 kb/s] speech encoding standard in step 3

Encrypt	Listening Results
All blocks	4
1 block-to-2 blocks	4
1 block-to-3 blocks	4
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

6.2.4 Encryption of G.726 [32 kb/s] encoded voice data

In the following, the change of encryption time of G.726 [32 kb/s] encoded voice data is also studied as well as the experiments in section 6.2.1. And then, the file is decoded before listening to the speech using the Sound Forge 4.5h software.

Nevertheless, there is a difference in terms of the preparatory process as well. That is, encoding the recorded file of 4,800,000 bytes with speech encoding algorithm, the ITU-T G.726 standard at 32 kb/s and saving the new file size of 2,400,000 bytes. The details of the experimental results appear in Tables 6.10, 6.11, and 6.12.

Table 6.10 Encryption and decryption times of G.726 [32 kb/s] speech encoding standard

Encrypt / Decrypt	Encryption Time (seconds)	Decryption Time (seconds)
All blocks	0.9	0.91
1 block-to-2 blocks	0.48	0.49
1 block-to-3 blocks	0.38	0.37
1 block-to-6 blocks	0.26	0.27
1 block-to-11 blocks	0.21	0.22

Table 6.11 The listening results of encryptions for G.726 [32 kb/s] speech encoding standard in step 2

Encrypt	Listening Results
All blocks	1
1 block-to-2 blocks	1
1 block-to-3 blocks	1
1 block-to-6 blocks	1
1 block-to-11 blocks	1

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

Table 6.12 The listening results of encryptions for G.726 [32 kb/s] speech encoding standard in step 3

Encrypt	Listening Results
All blocks	4
1 block-to-2 blocks	4
1 block-to-3 blocks	4
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

6.2.5 Encryption of AMR [12.2 kb/s] encoded voice data

And finally, the change of encryption time of AMR [12.2 kb/s] encoded voice data is analyzed, based on the experiments in section 6.2.1. However, in terms of the preparatory process, the recorded file of 4,800,000 bytes is encoded with speech encoding algorithm, the ETSI AMR standard at 12.2 kb/s, obtaining the new file size of 465,000 bytes. The experimental results are shown in Tables 6.13, 6.14, and 6.15.

Table 6.13 Encryption and decryption times of AMR [12.2 kb/s] speech encoding standard

Encrypt / Decrypt	Encryption Time (seconds)	Decryption Time (seconds)
All blocks	0.16	0.15
1 block-to-2 blocks	0.09	0.09
1 block-to-3 blocks	0.07	0.07
1 block-to-6 blocks	0.05	0.05
1 block-to-11 blocks	0.04	0.04

Table 6.14 The listening results of encryptions for AMR [12.2 kb/s] speech encoding standard in step 2

Encrypt	Listening Results
All blocks	1
1 block-to-2 blocks	1
1 block-to-3 blocks	1
1 block-to-6 blocks	1
1 block-to-11 blocks	2

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

Table 6.15 The listening results of encryptions for AMR [12.2 kb/s] speech encoding standard in step 3

Encrypt	Listening Results
All blocks	4
1 block-to-2 blocks	4
1 block-to-3 blocks	4
1 block-to-6 blocks	4
1 block-to-11 blocks	4

A 4-point scale: 4-fully understand, 3-fair, 2-poor, 1-no understanding at all

From the experiments in Section 6.2, the experimental results from encrypting with the G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards are compared as shown in Table 6.16.

Table 6.16 The comparison of listening results, encryption and decryption times, and the ratio of the before-encoding and after-encoding file sizes of G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards.

Encrypt / Decrypt	G.729 (Ratio : 16:1)			G.723.1 [6.3 kb/s] (Ratio : 20:1)			G.723.1 [5.3 kb/s] (Ratio : 24:1)			G.726 [32 kb/s] (Ratio : 2:1)			AMR [12.2 kb/s] (Ratio : 10:1)		
	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)
All blocks	1	0.11	0.10	1	0.08	0.08	1	0.07	0.07	1	0.9	0.91	1	0.16	0.15
1 block-to-2 blocks	1	0.06	0.05	1	0.05	0.04	1	0.04	0.04	1	0.48	0.49	1	0.09	0.09
1 block-to-3 blocks	**1**	**0.05**	**0.05**	**1**	**0.03**	**0.03**	**1**	**0.03**	**0.03**	**1**	**0.38**	**0.37**	**1**	**0.07**	**0.07**
1 block-to-6 blocks	4	0.02	0.02	2	0.02	0.02	2	0.02	0.02	1	0.26	0.27	1	0.05	0.05
1 block-to-11 blocks	4	0.02	0.02	3	0.02	0.02	3	0.01	0.01	1	0.21	0.22	2	0.04	0.04

Result Interpretation

To sum up, from Tables 6.1, 6.4, 6.7, 6.10, and 6.13, the more the unencrypted and the undecrypted blocks are, the less the encryption and the decryption times will be. Besides, in terms of eavesdropping simulation, it could be concluded that the more the number of unencrypted blocks is, the less voice security will be.

When we consider the results in Table 6.16, we find the interesting issues as follows. 1-to-3 blocks encryption is suitable for all speech encoding standards in this research. This is because it takes less time in encrypting and decrypting. However, it can achieve the data security, not being listened after encryption.

6.3 Information Tapping

Figure 6.3 Information tapping over the network

In the simulation, the experiments of transmitting voice data over the network and the simulation of information tapping have been employed as presented in Figure 6.3. Also, the 1-to-3 blocks encryption scheme, the finding from the previous experiment, is used in this experiment. Briefly speaking, in this experiment, we use one computer in the simulated situation, acting as two mobile phones: the sending and the receiving devices. The key is then sent from the sending device via SMS to the receiving device in the different signal channel from the signal channel of the conversation to prevent the key from being used in information tapping. After that,

the conversation between user A and user B begins. Encryption and information tapping are illustrated in Figure 6.4.

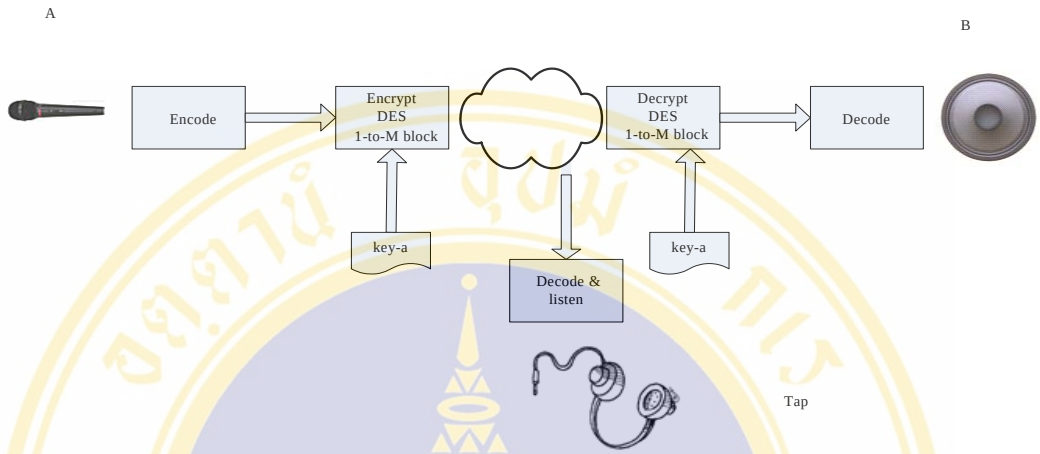

Figure 6.4 Diagram of conversation and information tapping among the network

As mentioned earlier, the experiments can be divided into three processes as follows.

Step 1

At first, exchange the key between the communicating parties: user A and user B, starting with the sending device (by user A) sends key-a to user B's receiving device. Next, user B's receiving device records the received key in the phonebook that is user A's (the sender's). In the same way, user B's sending device sends key-b to user A's receiving device. As the receiver, user A's receiving device also records the received key (key-b) in the phonebook that is user B's (the sender's).

Step 2

User A speaks via a microphone for about a minute. The speech is transformed into G.729 encoded voice data and encrypted using the DES algorithm, the 1-to-3 blocks encryption, with a key-a of 64 bits. Then, the data are transmitted over the network to user B. After that, the data are decrypted using the former key-a in the key exchange process and decoded into voice data as well as transformed to speech signal. The listening result of user A shows that it can be understood. Finally, do the same experiment, but change to the G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s],

and AMR [12.2 kb/s] speech encoding standards. Also, the results show that this can be understood.

Step 3

In the actual experiment, we do not intercept directly from the network, but we set up the simulation of information tapping before the data have been decrypted. Decode the data transmitted to user B but not decrypted yet, in the same way as encoding and listen to it. The listening results of the G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards indicate that it cannot be understood.

From the above-all experiments, we summarize our experimental results as follows. The 1-to-3 blocks encryption can be applied to all encryption schemes in this experiment, that is, the G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards. This is because it takes less time in the encryption and decryption processes as well as achieving data security. By this way, the approach can be extended to the VoIP standard or other mobile, portable devices such as the mobile phones. It could save battery power and meanwhile leverage the existing resources, that is, CPU (Central Processing Unit) power of the mobile phones, to do other tasks during the conversation instead of using in full encryption and decryption schemes.

CHAPTER VII
DISCUSSION AND CONCLUSION

7.1 Discussion

In this chapter, we discuss the experimental results presented in the previous chapter. The efficiency of the developed eavesdropping protection, as well as the way to further implement to other fields, is also discussed in this chapter.

7.1.1 Accuracy

The following, we discuss the accuracy of the experimental results presented in the previous chapter by considering the results in Table 7.1. By this way, the encryption and decryption times have been taken into consideration. From the table, when we change the option of encryption to 1-to-M block and M value is increased, this means the encrypted data are decreased. In other words, the first block of each packet is encrypted, while M-1 block is unencrypted, alternating like this until the end of the file. From this, the encryption and decryption times are decreased. Also, most of the experimental results correspond to the program logic. Nevertheless, there is something quite noticeable to mention. Some data do not correspond to the logic. For instance, the 1-to-11 blocks encryption of G.729 encoded voice data should take less encryption time than 1-to-6 blocks encryption. But in fact, the results are the same. At this point, it could be possible that the difference of the encryption time is not shown by the program in case of more than two decimal numbers. This is because the program shows only two decimal numbers or else this is because of other uncontrollable factors of the system in the experiment.

Table 7.1 The comparison of listening results, encryption and decryption times, and the ratio of the before-encoding and after-encoding file sizes of G.729, G.723.1 [6.3 kb/s], G.723.1 [5.3 kb/s], G.726 [32 kb/s], and AMR [12.2 kb/s] speech encoding standards.

Encrypt / Decrypt	G.729 (Ratio : 16:1)			G.723.1 [6.3 kb/s] (Ratio : 20:1)			G.723.1 [5.3 kb/s] (Ratio : 24:1)			G.726 [32 kb/s] (Ratio : 2:1)			AMR [12.2 kb/s] (Ratio : 10:1)		
	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)	Speech Quality	Encryption Time (sec.)	Decryption Time (sec.)
All blocks	1	0.11	0.10	1	0.08	0.08	1	0.07	0.07	1	0.9	0.91	1	0.16	0.15
1 block-to-2 blocks	1	0.06	0.05	1	0.05	0.04	1	0.04	0.04	1	0.48	0.49	1	0.09	0.09
1 block-to-3 blocks	1	0.05	0.05	1	0.03	0.03	1	0.03	0.03	1	0.38	0.37	1	0.07	0.07
1 block-to-6 blocks	4	**0.02**	0.02	2	0.02	0.02	2	0.02	0.02	1	0.26	0.27	1	0.05	0.05
1 block-to-11 blocks	4	**0.02**	0.02	3	0.02	0.02	3	0.01	0.01	1	0.21	0.22	2	0.04	0.04

7.1.2 Performance of the system

In the following section, we discuss some issues in terms of its performance. From the developed eavesdropping protection, we are satisfied with the experimental results and the operation of system prototype that meet the set objective. However, in case that we apply the system to the real applications, further research should be done in the related fields, for example, the implementation to the real-time conversation systems.

Moreover, in the real applications, we may use other algorithms according to circumstances other than the DES algorithm that is used in this experiment. In this case, the efficiency of the CPU or the sending and receiving devices' resource should be considered, especially when we use with mobile, portable devices that have the limitation in this mater.

7.1.3 Application of voice data transformation system

In this section, we discuss the application of voice data transformation system. Based on the experimental results, this is the approach to prevent eavesdropping. In this case, the business owners can apply the voice data transformation system to many fields such as mobile phones, VoIP, and so on. Besides, in the near future, the integration of the mobile phones and the Internet leads to the new standard. Therefore, we recommend that the eavesdropping protection that provides the real-privacy circumstance as presented in this research should be another concern.

7.1.4 Program complexity

In our developed programs in the experiment, we use TCP/IP Socket to send and receive the data over the network system. And we use API (an Application Programming Interface) module in the DES encryption algorithm, running on Linux TLE 5.0, which is different from the prototype programs of the system developed by

Winsock Library. Also, the API module used in the DES encryption algorithm, running on Windows Me is employed in this experiment.

The programs are developed by the C language and are not complicated, so it could be easily ported to develop under other operating systems in the future.

7.2 Conclusion

According to the proposed approach of voice eavesdropping protection over digital networks, it can meet the defined objective as appeared in the experimental results and the system prototype. So, it can be concluded that this approach is indeed another effective choice of protection against eavesdropping.

REFERENCES

1. A. Servetti, J. C. De Martin. Perception-Based Partial Encryption of Compressed Speech. IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 8, November 2002.
2. ITU-T G.729. Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP), March 1996.
3. J.-I. Guo, J.-C. Yen, H.-F. Pai. New Voice over Internet Protocol Technique with Hierarchical Data Security Protection. IEE Proceedings-Vision, Image and Signal Processing, Vol. 149, No. 4, August 2002.
4. M. Baugher, et al. The Secure Real-time Transport Protocol (SRTP). IETF, RFC 3711, March 2004.
5. R. Blom, E. Carrara, F. Lindholm, K. Norrman, M. Näslund. Conversational IP Multimedia Security. IEEE Mobile and Wireless Communications Network, 2002.
6. A. Mehrotra, L. S. Golding. Mobility and Security Management in the GSM System and Some Proposed Future Improvements. Proceedings of the IEEE, Vol. 86, No. 7, July 1998.
7. Security Related Network Functions (Release 6), 3GPP TS 43.020 V6.1.0, December 2004.
8. A. B. Rekha, B. Umadevi, Y. Solanke, S. R. Kolli. End-to-End Security for GSM Users. IEEE ICPWC 2005.
9. ITU-T G.723.1. Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s, March 1996.
10. ITU-T G.107. The E-Model, a Computational Model for Use in Transmission Planning, March 2005.
11. A. P. Markopoulou, F. A. Tobagi, M. J. Karam. Assessment of VoIP Quality over Internet Backbones. IEEE INFOCOM, 2002.
12. ITU-T G.113. Transmission Impairments due to Speech Processing, February 2001.

13. ITU-T G.113 Appendix I. Appendix I: Provisional Planning Values for the Equipment Impairment Factor I_e and Packet-Loss Robustness Factor B_{pl}, May 2002.
14. ITU-T G.726. 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), December 1990.
15. Adaptive Multi-Rate (AMR) Speech Codec Transcoding Functions (Release 6), 3GPP TS 26.090 V6.0.0, December 2004.

APPENDIX

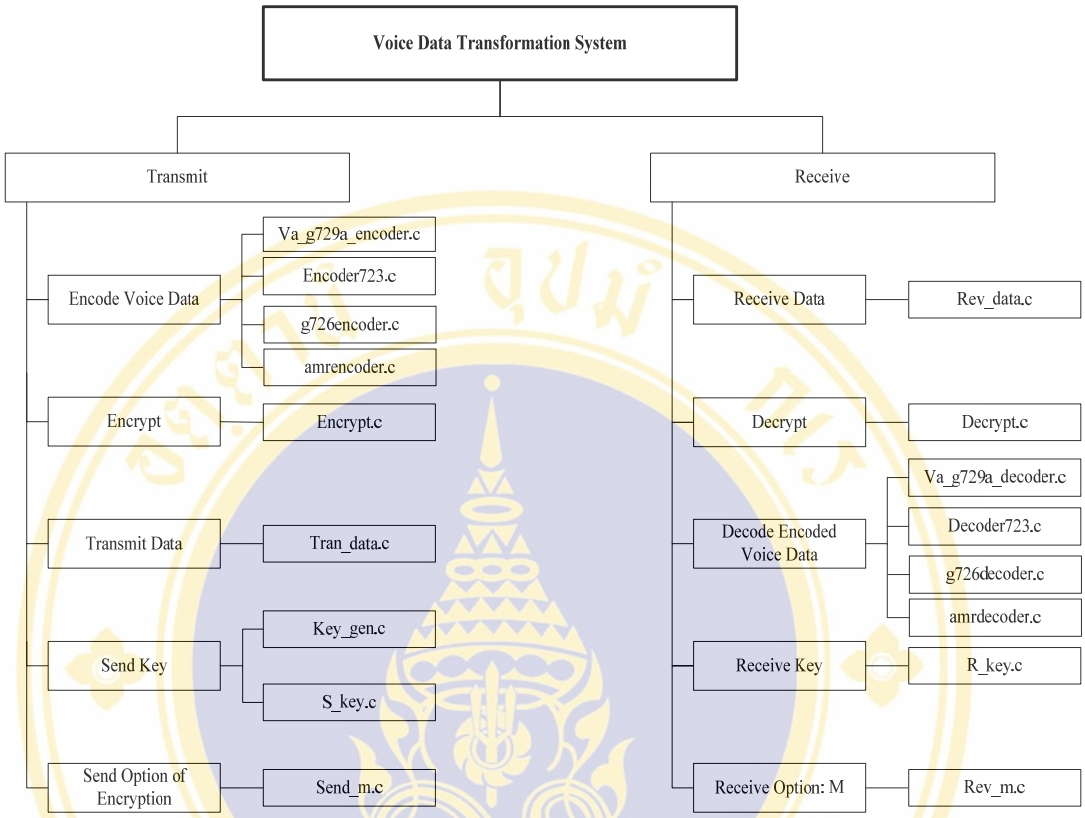

Figure A.1 Program modules of voice data transformation system

As for the program modules of this system, the descriptions of them are as follows.

1. Va_g729a_encoder.c: the program that is used to encode the voice data with the G.729 standard
2. Encoder723.c: the program that is used to encode the voice data with the G.723.1 standard
3. g726encoder.c: the program that is used to encode the voice data with the G.726 standard
4. amrencoder.c: the program that is used to encode the voice data with the AMR standard
5. Encrypt.c: encrypt the encoded voice data
6. Tran_data.c: transmit the data over the network
7. Key_gen.c: generate the key used in encryption

8. S_key.c: send the key to the communicating parties
9. Send_m.c: send the option used in encryption
10. Rev_data.c: receive the data from the network
11. Decrypt.c: decrypt the voice data
12. Va_g729a_decoder.c: decode with the G.729 standard to obtain the original voice data
13. Decoder723.c: decode with the G.723.1 standard to obtain the original voice data
14. g726decoder.c: decode with the G.726 standard to obtain the original voice data
15. amrdecoder.c: decode with the AMR standard to obtain the original voice data
16. R_key.c: receive the key from the speaker
17. Rev_m.c: receive the option: M

Program Coding

The following is a part of program coding used in this research.

```
/*      va_g729a_encoder.c  */


#include "stdio.h"
#include "va_g729a.h"

void main(int argc, char *argv[])
{
        int n_frame;
        FILE* f_in;
        FILE* f_out;
        short                   speech[L_FRAME];
        unsigned char   serial[L_FRAME_COMPRESSED];

        if (argc != 3)
        {
                printf("Usage: %s infile outfile\n", argv[0]);
                return;
        }

        if ( (f_in = fopen(argv[1], "rb")) == NULL)
        {
                printf("\nError opening input file %s!", argv[1]);
                return;
        }

        if ( (f_out = fopen(argv[2], "wb")) == NULL)
        {
                printf("\nError opening output file %s!", argv[2]);
                return;
        }

        va_g729a_init_encoder();
        n_frame = 0;
        while (fread(speech, sizeof(short), L_FRAME, f_in) == L_FRAME)
        {
                printf("Encode frame %d\r", ++n_frame);
                fwrite(serial, sizeof(char), L_FRAME_COMPRESSED, f_out);
        }

        fclose(f_out);
        fclose(f_in);
}
/* end of file: va_g729a_encoder.c  */
```

```
/*     va_g729a_decoder.c       */


#include "stdio.h"
#include "va_g729a.h"

void main(int argc, char *argv[])
{

        int n_frame;
        FILE* f_in;
        FILE* f_out;
        unsigned char   serial[L_FRAME_COMPRESSED];
        short                   synth[L_FRAME];
        int                     bfi;

        if (argc != 3)
        {
                printf("Usage: %s infile outfile\n", argv[0]);
                return;
        }

        if ( (f_in = fopen(argv[1], "rb")) == NULL)
        {
                printf("\nError opening input file %s!", argv[1]);
                return;
        }

        if ( (f_out = fopen(argv[2], "wb")) == NULL)
        {
                printf("\nError opening output file %s!", argv[2]);
                return;
        }
        va_g729a_init_decoder();
        n_frame = 0;
        while (fread(serial, sizeof(char), L_FRAME_COMPRESSED, f_in) ==
L_FRAME_COMPRESSED)
        {

                printf("Decode frame %d\r", ++n_frame);
                bfi = 0;
                va_g729a_decoder(serial, synth, bfi);
                fwrite(synth, sizeof(short), L_FRAME, f_out);
        }

        fclose(f_out);
        fclose(f_in);

}
/* end of file: va_g729a_decoder.c  */
```

```
/* key_gen.c */

#include <sys/types.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h>


char key[8];
int  i,tmp[8];
main()
{
            rdom();
            savekey();

}
rdom()
{
            int x,y,z,loop;
            double result;
            srand( (unsigned)time( NULL ) );
            for( i = 0;   i < 8;i++ )
            {
                        tmp[i] = rand();
            }

            for (z=0;z<8;z++)
            {
                        /*initstate(); */
                        x = tmp[z];
                        y = 1+(int) (128.0*x/(RAND_MAX+1.0));
                        key[z] = y;
                        }
            return 0;
}
savekey()

{
FILE  *aaa;
            aaa = fopen("keys_ab.key","wb");
            fwrite(key,1,8,aaa);
            fclose(aaa);
            return 0;
}
/* end of file: key_gen.c  */
```

```
/* encrypt.c */


#include <stdio.h>
#include <stdlib.h>
#include "QuickCrypt.h"
char keys[8];
FILE *stream, *stream1,*stream2,*st3;
int  j,k,numread,m;
char cm[8];
char *s;
main()
{
        if((st3 = fopen("keys_ab.key","rb"))!=NULL)
        {
                numread= fread(keys,1,8,st3);
        }
        fclose(st3);

        if((stream2 = fopen("options.out","rb"))!=NULL)
        {
                numread= fread(cm,1,8,stream2);
        }
        printf("st1\n");
        s = cm;
        m = atoi(s);
        fclose(stream2);
        if((stream = fopen("fread.out","rb")) != NULL)
        {
                if( (stream1 = fopen("fread1.out", "wb")) !=NULL)
                {
                        j=1;
                        k=0;
                        doencrypt();
                }
        }

fclose(stream);
fclose(stream1);
}

doencrypt()
{
        char work[8];
        int cnt;
        unsigned char context[SLC_DES_CONTEXTSIZE];
        SL_DES_Init(context,0,keys,SLC_DES_DEFAULTKEYSIZE);
        for(;;)
                {
                if((cnt = fread(work,1,8,stream)) != 8)
                        {
                                work[7] = cnt;
```

```
                        }

                if (j > k)
                        {
                                /*endes(ks,work);          /* Encrypt block */
                                SL_DES_ProcessBlock(context,work,work);
                                k=k+m;

                        }
                        j++;
                        fwrite(work,1,8,stream1);
                if(cnt != 8)
                        break;
                }
return 0;
}
/* end of file: encrypt.c */
```

```
/* decrypt.c  */

#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include "QuickCrypt.h"
char keyr[8];
FILE *stream, *stream1,*stream2,*st3;
int  j,k,m,numread;
char *s;
char cm[8];
main()
{
        if((st3 = fopen("keyr_ab.key","rb"))!=NULL)
        {
                numread= fread(keyr,1,8,st3);
        }
        fclose(st3);
        if((stream2 = fopen("optionr.out","rb"))!=NULL)
        {
                numread= fread(cm,1,8,stream2);
        }
        s = cm;
        m = atoi(s);
        fclose(stream2);
        if((stream = fopen("fread1.out","rb")) != NULL)
        {
                if( (stream1 = fopen("fread2.out", "wb")) !=NULL)
                {
                        j=1;
                        k=0;
                        dodecrypt();
                }
        }

fclose(stream);
fclose(stream1);
return 0;
}

dodecrypt()

{
        char work[8],nwork[8];
        int cnt;
        unsigned char context[SLC_DES_CONTEXTSIZE];
        SL_DES_Init(context,1,keyr,SLC_DES_DEFAULTKEYSIZE);
        cnt = fread(work,1,8,stream);     /* Prime the pump */
        for(;;){
                if (j >k )
                {
                 SL_DES_ProcessBlock(context,work,work);
```

```
                    /*dedes(ks,work);*/
                    k=k+m;
                   }
                    j++;
                    memcpy(nwork,work,8);
                   /* Try to read next block */
                   cnt = fread(work,1,8,stream);
                   if(cnt != 8)
                   {          /* Can "only" be 0 if not 8 */
                    cnt = nwork[7];
                    if(cnt < 0 || cnt > 7)
                    {
                     fprintf(stderr,"Corrupted file or wrong key\n");
                    }
                    else if(cnt != 0)
                              fwrite(nwork,1,cnt,stream1);
                              exit(0);
                   }          else
                   {
                              /* Now okay to write previous buffer */
                              fwrite(nwork,1,8,stream1);
                              }

                   }
}
/* end of file: decrypt.c */
```

BIOGRAPHY

NAME	Mr. Sathaporn Kassuvan
DATE OF BIRTH	30 July 1970
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Mahanakorn University of Technology, 1995 : Bachelor of Industrial Technology (Electrical Engineering) Mahidol University, 2006 : Master of Science (Computer Science)
POSITION&OFFICE	Ascon Construction PCL. 1768 Thai Summit Tower, 25th Floor, New Petchburi Road, Khwaeng Bang Kapi, Khet Huay Khwang, Bangkok, Thailand, 10310 Position : IT Manager Tel. 02-652-8999 E-mail: kengmeta@yahoo.com
HOME ADDRESS	530/326 Moo 1, Tambon Phraekkasa Mai, Amphur Muang, Samutprakan, Thailand, 10280 Tel. 02-3343092