การเข้ารหัสลับแบบเลือกได้สำหรับข้อมูลเสียงที่ถูกบีบอัด

พิชิต ทนันชัย [1]
มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา 128 ถ.ห้วยแก้ว อ.เมือง จ.เชียงใหม่ 50300
ธำรงรัตน์ อมรรักษา [2]
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี บางมด ทุ่งครุ กรุงเทพฯ 10140

บทคัดย่อ

เพื่อทำให้การส่งข้อมูลเสียงผ่านเครือข่ายแบบเวลาจริงประสบผล จึงได้นำเสนอวิธีการเข้ารหัสลับแบบเลือกได้สำหรับข้อมูลเสียงที่ถูกบีบอัดในบทความนี้ นั่นคือ แทนที่จะเข้ารหัสลับข้อมูลเสียงโดยตรงทั้งหมด วิธีการที่นำเสนอจะเข้ารหัสลับเพียงแค่ส่วนที่สำคัญขนาดเล็กที่ดึงออกมาจากข้อมูลเสียงนั้นๆ แล้วส่วนที่ถูกรหัสลับดังกล่าวจะนำมาใช้ป้องกันข้อมูลเสียงในส่วนที่เหลือ โดยการพิจารณาข้อมูลเสียงแบบ MPEG 1 Layer III พารามิเตอร์บางตัวสามารถนำมาใช้ในฐานะส่วนที่สำคัญ ดังนั้นในบทความนี้ ได้พิจารณาพารามิเตอร์ 4 ตัวที่ดึงออกมาจากข้อมูลข้างเคียงของข้อมูลเสียงแบบเอ็มพีสาม ซึ่งมีชื่อว่า *main__data__begin, scfsi, part2__3__length* และ *table__select* สำหรับการเข้ารหัสลับแบบเลือกได้ ประสิทธิภาพที่ได้ เมื่อนำไปประยุกต์ใช้ในขั้นตอนการเข้ารหัสลับ ในเชิงคุณภาพของเสียงที่ถูกเข้ารหัสลับ และความซับซ้อนของกระบวนการเข้ารหัสลับ จะถูกประเมินและเปรียบเทียบกับขั้นตอนการเข้ารหัสลับแบบทั่วๆ ไป ผลการทดลองแสดงให้เห็นว่า การเข้ารหัสลับที่ *main__data__begin* จะช่วยลดช่วงเวลาที่ใช้ในการเข้ารหัสลับได้ร้อยละ 99.71 โดยเฉลี่ย ขณะที่ยังคงรักษาระดับความปลอดภัยของข้อมูลที่ถูกเข้ารหัสลับไว้ได้ วิธีการดังกล่าวช่วยให้การส่งเพลง แบบเอ็มพีสามที่เข้ารหัสลับแบบเวลาจริงผ่านอินเตอร์เน็ตเป็นไปได้ นอกจากนี้ยังสามารถนำไปประยุกต์ใช้กับมาตรฐานการบีบอัดข้อมูลเสียงแบบอื่นๆ ได้ด้วย

---

[1] *อาจารย์ แผนกวิชาเทคนิคคอมพิวเตอร์*

[2] *รองศาสตราจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์*

Selective Encryption for Compressed Audio

Pichit Tananchai [1]
Rajamangala University of Technology Lanna, Muang, Chiang Mai 50300
Thumrongrat Amornraksa [2]
King Mongkut's University of Technology Thonburi, Bangmod, Toongkru, Bangkok 10140

Abstract

To achieve a real time secure transmission of audio over networks, a selective encryption method for compressed audio is proposed in this paper. That is, instead of directly encrypting the entire audio, the proposed method only encrypts a small important part extracted from that audio, and the encrypted part is used to protect the contents of the remaining audio. By considering a standard MPEG 1 Layer III bit-stream, some parameters can be extracted and used as the important parts. Thus, in this paper, we consider four parameters extracted from the side information of a MP3 bit-stream, namely, *main_data_begin*, *scfsi*, *part2_3_length* and *table_select*, for the selective encryption. Their performances, when implemented in an encryption scheme, in terms of quality of the encrypted audio and complexity of the encryption process, are evaluated and compared with ordinary encryption methods. The results show that, by using the *main_data_begin* as an important part, the processing time required in the encryption process can be reduced by 99.71 %, on average, while maintaining the security level of the encrypted audio. Our approach enables a real time encryption for MP3 music delivery via the Internet. Furthermore, it can be applied to other audio compression standards.

[1] *Lecturer, Technique Computer Division.*

[2] *Associate Profersor, Department of Computer Engineering.*

1. Introduction

Nowadays, advances of Internet technology have improved access to digital information. The widening Internet bandwidth and availability of digital consumer electronics recording and storage devices have increased the demand for multimedia services. Multimedia data in digital form can be reproduced infinitely without any loss of quality. Everyone is now aware of the real dangers of inaccurate personal information and uncontrolled access to files. The security of multimedia data in digital distribution networks is commonly provided via mathematical processes, which can be simply achieved by transforming the multimedia data into cipher text, non-intelligent version of the multimedia data, to limit access to authorized users. Such process, known as encryption, requires certain amount of processing time, and slows down the entire system when applied to a large multimedia data, especially with a compressed audio.

Multimedia security is thus an important issue in compressed audio applications, such as, real time audio on demand on the Internet or satellite broadcasting, and audio media delivery and audio media link via ISDN. To reduce the processing time required in the encryption process, the *important part* of the multimedia data is extracted, encrypted and used to protect the contents of the remaining. This method is well known as selective encryption. For instance, by considering the I-frames in a MPEG video bit-streams as the important parts, the I-frames can be encrypted only and the results are used the protect the remaining video [1]. When the encrypted video is decoded without decrypting first, the coding error will propagate to P-frames and B-frames; so these frames, too, will be in corrupted form. Although the decoded video sometimes reveals a certain amount of information e.g. general contents of the video, it is still considered "secure enough" for most multimedia applications, where the requirement is merely to destroy the commercial value of the source material.

For the security of compressed audio, Thorwirth et al [2] proposed a secure method for MP3 music transmission to enable Internet based audio delivery and protection of digitized music against illegal distribution. In their approach, the encryption was applied to layers of audio quality associated with compressed frequency based main audio, without compromising the stream bit-rate or the network bandwidth requirement. By considering header information describing the structure of compressed audio, exacts frequency spectrum boundaries were determined and used for encryption. However, their method required a preprocessing time for reading the header information and determining the boundaries that represent layers of audio quality to be encrypted. Later, the perceptual based approach for MP3 encryption was proposed by Torrubia and Mora [3]. In their approach, the Huffman code bits were modified in a way that the decoder could build the corresponding 576 frequency lines. Each Huffman code word was then substituted with another code word of the same size, and encrypted by XORing the results with a pseudo random bit-stream. Nevertheless, there was lacked of security investigation against attacks in their encryption scheme, especially against the brute force one.

A systematic online music protection for MP3 bit-stream was also proposed by Gang et al [4]. Basically, their approach provided three levels of protection. Level one was "slight" protection, where the protected bit-stream presented a satisfactory music quality for a casual listener, but not good enough for Hi-Fi reproduction. Level two was

"moderate" protection, where the protected content is meaningful and the main music features are kept, but with obvious degradation. Level three was "maximum" protection, where the music content is completely destroyed thus rendering the MP3 bit-stream meaningless. However, their approach required long period of encryption time and was quite impractical. Among the earliest work, a frequency-selective partial encryption of compressed audio was proposed by Servetti et al [5], where the encryption scheme employed low pass filters in the compressed domain to limit the frequency content of audio materiel. In their scheme, the resulting bit-stream could be decoded without error by any MP3 standard decoder. However, the high frequency components, at approximately 1.5 to 5.5 kHz, of the audio signal would be rejected and ignored. Using this technique, the compressed audio was not entirely protected, and listeners could still listen to the unprotected parts in other frequencies. In some circumstances where a real time secure transmission of audio over networks e.g. a music delivery via the Internet is required, the encrypted methods previously mentioned are not suited since their complexity in encryption process is still too high for real time applications. To achieve that, a fast and effective encryption algorithm must be developed.

In this paper, an alternative approach based on selective encryption is proposed for protecting the contents of compressed audio, especially for the MPEG-1 Layer III standard. Conceptually, a small important part within the MP3 bit-stream is extracted and encrypted. To obtain the original MP3 bit-stream, one needs to decrypt that part before entering MP3 decoder. Otherwise, coding errors will propagate to other parts of the resultant decoded bit-stream, and make the whole thing corrupted. Ultimately, the decoded audio should be unintelligible to human being. In the next section, background of MP3 audio coding standard is given. Four parameters extracted from the MP3 bit-streams are considered and discussed in Section 3. Section 4 describes details of the experiments and the evaluation methods. The results are shown and discussed in Section 5. Conclusions on this work are finally drawn in Section 6.

2. Fundamental Concept

Basically, MPEG-1 Layer III audio bit-stream, known as MP3, is achieved based on perceptual modeling without substantial quality loss once decoded. It has ability to preserve near CD quality music. MP3 is standardized under ISO/IEC International Standard IS 11172-3 [6], widely known as Moving Picture Experts Group (MPEG). Practically, MP3 audio is built up from several smaller parts, called frames. There are independent items in that each frame has its own header and audio information. Moreover, the MP3 audio has no main header so that its parts can be cut and played back independently and correctly, without the need of the whole file (this should be done on frame boundaries but most applications will handle incorrect headers).

2.1 Structure of MP3

As shown in Fig. 1, each frame header consists of 4 bytes information containing a synchronization word together with a description of the frame e.g. bit-rate, mono or stereo mode, and etc. The synchronization word, found at the beginning of each frame, enables MP3 decoder to lock onto the signal at any point of the bit-stream. Cyclic Redundancy Code (CRC) makes it possible to check the most sensitive data; the header and side information, for transmission error. Those two parameters are considered as the most sensitive data

because if they are missing or incorrect, the whole frame will be corrupted. Next, side information consists of information needed to decode the main data, while main data consists of the contents the coded audio. It is worth noting that errors in the main data will only distort a part of that frame. The last parameter is ancillary data, which is optional and ignored by decoder. It is designed for inserting user defined data into the bitstream, e.g. song title etc.

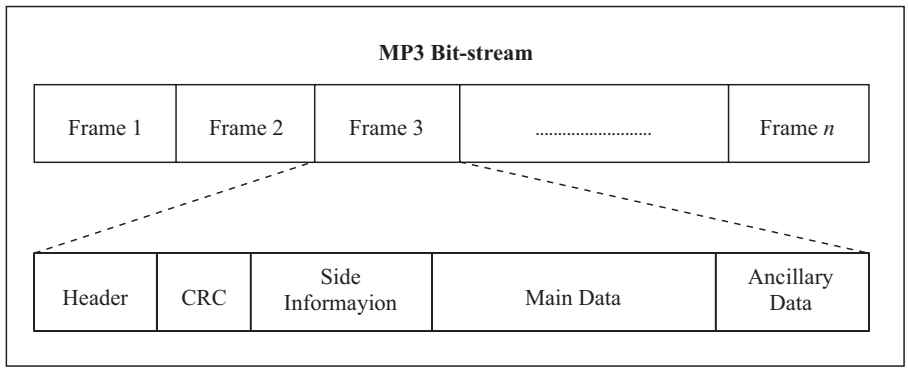

Fig. 1 Structure of MP3 audio

2.2 Decoding of MP3

Block diagram of MP3 decoder is shown in Fig. 2. Firstly, when the sync and error checking block receives the incoming MP3 bit-stream, every frame within that bit-stream must be identified by searching for a synchronization word. Note that it is not possible for this block to extract the correct information needed, if no frames are located. The side info decoding block will then decode the encoded side information, needed for Huffman decoding block to perform its task correctly. Finally, the last block will reconstruct the audio bit-stream in PCM format by using synthesis poly phase filter banks.

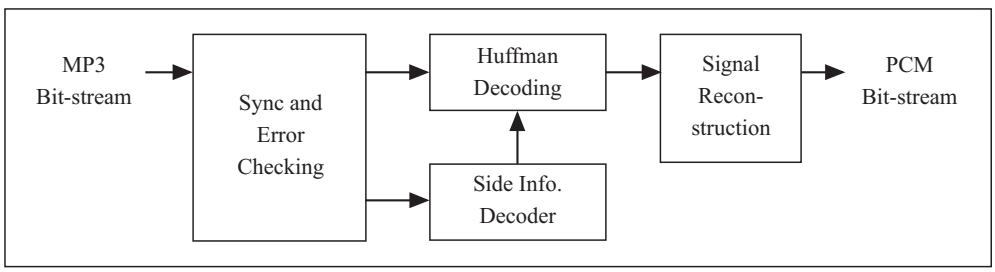

Fig. 2 Block diagram of MP3 decoder

3. Proposed Encryption Method

Conventionally, the encryption is directly applied to the entire MP3 data which requires a considerable amount of encryption time. Someone may try to apply the encryption to the main data part only. Its improved performance is however small and not obvious. Therefore, in this research work, the encryption is selectively applied to a part of the MP3 bit-stream. According to ISO/IEC 11172-3, we found that four parameters located in the side information, namely, *main_data_begin, scfsi*, *part2_3_length* and *table_select*, can be used as the important parts. Descriptions for these four parameters are given in Table 1.

Table 1 Descriptions of four chosen parameters

Parameters	Descriptions
main_data_begin	Contains information used to locate the first bit of main data within a frame, and a negative offset from the first byte of the audio synchronize word.
scfsi	Contains information used to select the scale factor
part2_3_length	Contains number of main data bits used for Huffman code bits
table_select	Contains a different Huffman code bits table used for quantization of the sample signal

The proposed method can be implemented in an encryption scheme as shown in Fig. 3. Principally, an incoming MP3 bit-stream is first divided into successive frames. The important part within the side information from each frame is then extracted and entered to the encryption process. Such encrypted parts are finally inserted back into the MP3 bit-stream at the same corresponding locations. The decryption process can be simply achieved by following the same steps with the replacement of a decrypting block.

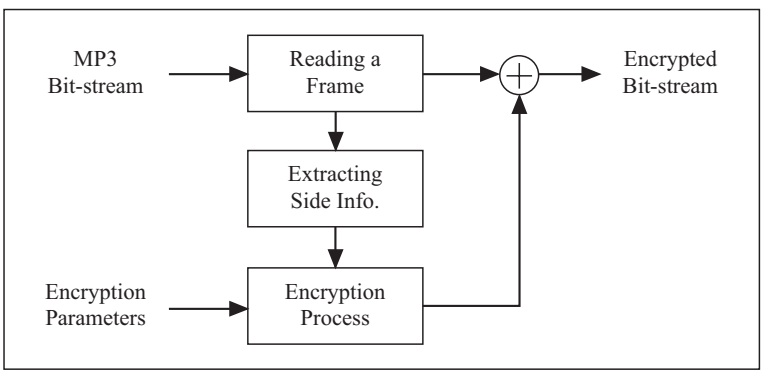

Fig. 3 Block diagram of the encryption scheme

In our encryption process, a well known RC4 [7, 8] algorithm was used. This algorithm has been used in many standards e.g. IEEE 802.11 in Wireless Encryption Protocol (WEP), and commercial software packages e.g. Lotus Notes and Oracle Secure SQL. The RC4 uses a variable length key from 1 to 256 bytes to initialize a 256-byte state table. The state table is used for subsequent generation of pseudo-random bytes, and to generate a pseudo-random stream which is later used for XORing with the important parts. Each element in the state table is swapped at least once. Because of export restrictions in the USA, its key in a world wide distributed code is limited to 40 bits (or 128 bits sometimes). However, the algorithm has the capability of using key lengths between 1 and 2,048 bits.

4. Experimental Settings

In all experiments, five output bit-rates and four different characteristics of MP3 compressed audio were tested. The bit-rates we selected were 32, 64, 128, 192 and 320 kbps, the characteristics of the testing audio are given in Table 2 and 3. Note that *Spme50_1* (male speech) is a standard test bit-stream under SQAM [9] which is normally used for human speech testing, while *Hecommon* is a standard test bit-stream under the MPEG Organization Standard [10]. River and *Humanity* are merely ordinary music.

Table 2 Characteristics of the testing audio

Audio	Length (s)	Characteristics
Spme50_1	17.92	- A human speech that contains middle frequency components as its background.
River	39.84	- A song with a singer that contains low, middle and high frequencies.
Humanity	39.84	- Pure music without a singer performing.
Hecommon	1.645	- A test pattern sine wave at 1 kHz and 0 dB that contains a single frequency throughout the entire bit-stream.

Table 3 Parameters size of the testing audio, sampled at 44.1 kHz

Bit-rate (kbps)	Frame (bytes)	Size (Kbytes)			
		Spme50_1	***River***	***Humanity***	***Hecommon***
32	104	71	156	157	7
64	208	141	312	313	14
128	417	282	624	625	27
192	626	423	936	937	41
320	1,044	705	1,560	1,560	66

To evaluate the quality of encrypted audio, both objective and subjective quality evaluations were performed. Signal-to-Noise Ratio (SNR) is the most popular objective based evaluation method, and is used in our experiments, to compare physical differences between the original and the encrypted audio. It employs mathematical models simulating the human observers, and is widely used as standard metric for audio quality evaluation. Fundamentally, the signal strength relative to background noise is measured, and the result is given as a ratio represented in decibels (dB). For instance, if the signal strength is equal to the noise strength, the output SNR will be 0 dB. In contract to the SNR, the subjective based methods evaluate the quality of audio signal by using the human ear. Usually, the output audio is played back to a number of listeners, and each listener will independently give a score in accordance to such audio quality. Average score is then computed and used to indicate how much the quality of the encrypted audio differs from the original. Referring to [9], six ranges of a given score are designed for the subjective quality evaluation in our listening tests. Detail of each score level is given in Table 4. Notice that the score of 5 gives the best performance in term of encryption i.e. the listeners cannot recognize anything from the played back encrypted audio. In the listening tests, various versions of encrypted audio were randomly played back to ten listeners, aging between 15 to 30 year olds. All of them were in good health and had no known hearing problems. For the complexity evaluation of the encryption process, the percentage of encrypted data is calculated (a ratio of the size of encrypted data over the size of entire audio, all multiplied by 100). The processing time required in the encryption process is also measured and compared.

Table 4 Subjective listening test score

Audio Quality	***Catastrophic***	***Very Annoying***	***Annoying***	***Slightly Annoying***	***Audible***	***Insensitive***
Score	5	4	3	2	1	0

The experiments to be carried out were divided into three parts. The first and second parts focused on the objective and subjective qualities of the testing audio, encrypted at various important parts i.e. *main_data_begin*, *scfsi*, *part2_3_length* and *table_select*, respectively, while the final part focused on the complexity of the encryption process implementing with different important parts. The results are finally plotted and compared with the ones obtained from ordinary encryption method i.e. by directly encrypting the main data.

5. Results and Discussions

According to the structure of MP3, the size of the chosen parameters within a frame is fixed across bit-rates. In contrary, the size of the main data varied from 84 to 1,024 bytes as the bit-rate varied from 32 to 320 kbps. This is because the MP3 encoder adds more redundancy into the main data at a higher bit-rate. The size of five parameters mentioned above at various bit-rates is summarized in Table 5.

Table 5 Amount of the encrypted parts per one frame

Bit-rate (kbps)	Encrypted part per one frame (byte)				
	main data	***main_data_begin***	***scfsi***	***part2_3_length***	***table_select***
32	84	1	1	2	2
64	188	1	1	2	2
128	397	1	1	2	2
192	606	1	1	2	2
320	1,024	1	1	2	2

5.1 Quality of the encrypted audio

The plots of SNR at various important parts, averaged from four testing bit-streams, are illustrated in Fig. 4. It can be seen that the result obtained from encrypting the *main_data_begin* was close to that obtained from encrypting the *main data*. It means encrypting the *main_data_begin* degrades the output audio quality as close as encrypting the main data. Note that a lower SNR indicates a better performance in concealing the contents of the audio. From the figure, the best and worst performances were obtained from the encryptions of the *main_data_begin* and the *scfsi*, respectively. Also, the curves are not linear since the errors propagated in the decoded audio were not proportionally varied to the size of the testing bit-streams.

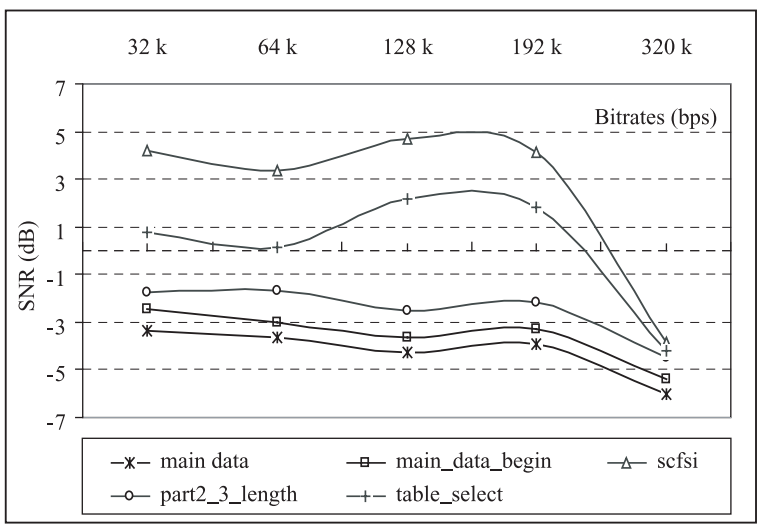

Fig. 4 Average SNR of the encrypted audio

For the subjective quality of the encrypted audio, the plots of the resulting listening test score averaged from four testing bit-streams are shown in Fig. 5. Similar to the objective evaluation results, the encryptions of the *main_data_begin* and the *scfsi* gave the best and worst performances, respectively.

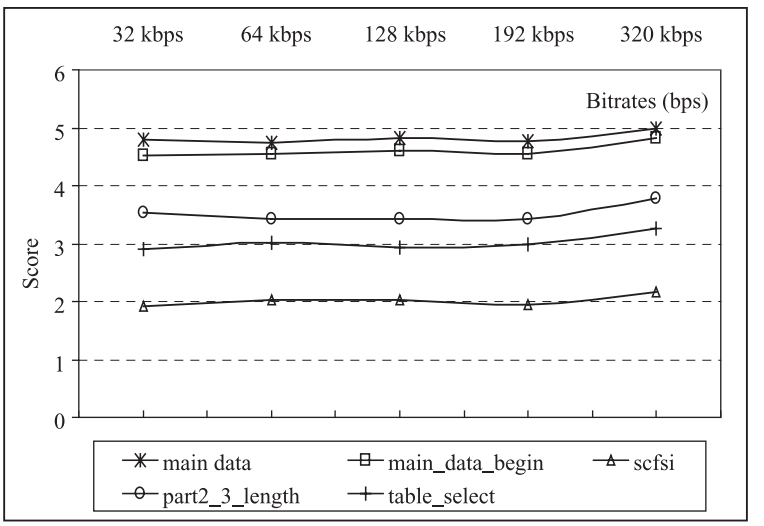

Fig. 5 Average listening test score of the encrypted audio

5.2 Complexity of the encryption process

The percentages of data encryption required at various important parts are illustrated in Fig. 6, while the average processing times required in the encryption process are shown in Fig 7. It is obvious from both figures that encrypting the main data was the most complex approach to protect the contents of the encrypted audio. As expected, encrypting the *main_data_begin* gave the lowest complexity, compared to the others. It should be noticed that while the audio output bit-rate was increasing, the percentage of the encrypted important parts was proportionally decreased, and the processing time required for encrypting the important parts was slightly increased. This is because, as mentioned earlier, the size of the important parts is fixed across bit-rates. It should also be noticed that the complexity of our encryption scheme is greatly lower than the encryption technique proposed in [3]. This is because, encrypting the Huffman code words is equivalent to encrypting the main data, based on the fact that most information contained in the main data are the Huffman code words. Various versions of improperly decoded audio waveform 'Spme50_1' at different important parts were shown in Fig. 8. Actually, they were decoded without decrypting first.

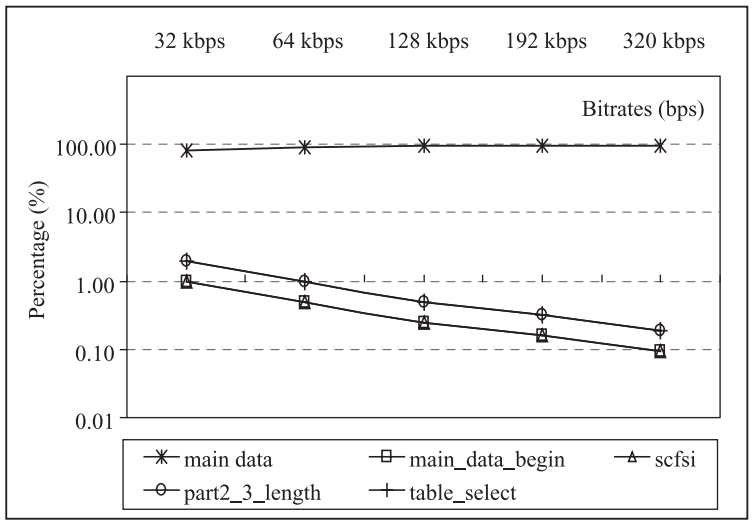

Fig. 6 Encryption percentages at different important parts

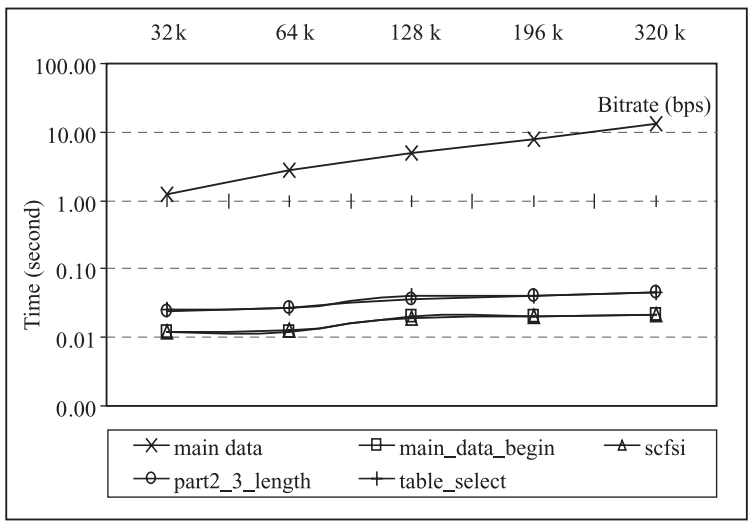

Fig. 7 Average processing time at different important parts

For the test of practical implementation, using our approach, the encryption time required for the MP3 file with the length of approximately 18 minutes was less than 0.1, while in [2], the encryption time required for the 10 minutes MP3 file was 0.68 seconds. Obviously, the encryption speed of the proposed method is more than twelve times faster than the method previously proposed in [2]. Note that the method proposed in [5] is not comparable since the encrypted result is merely a low quality version of the original music file.

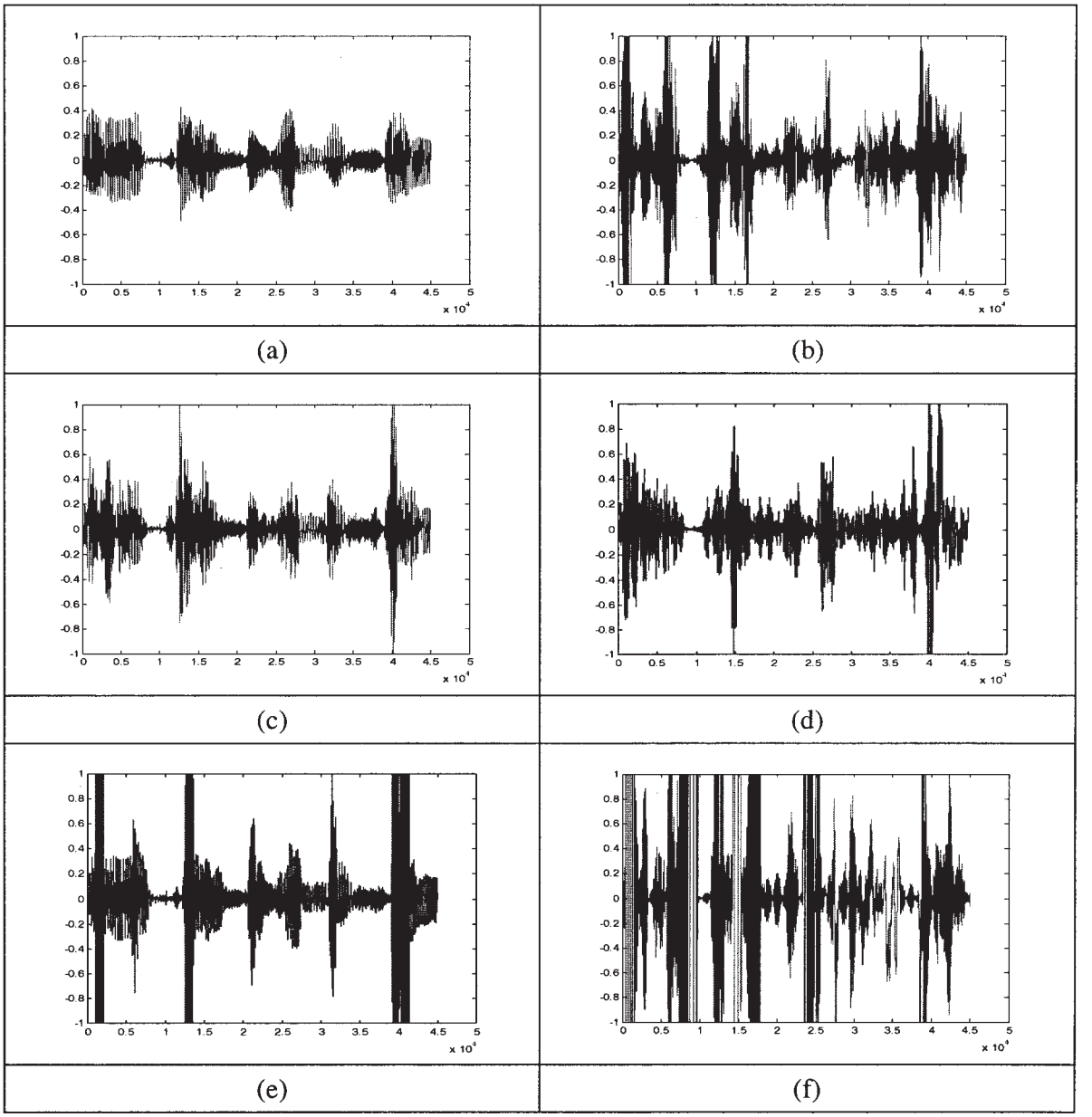

Fig. 8 (a) Original waveform '*Spme50_1*' sampled at 128 kbps (b) the improperly decoded versions encrypted at *main_data_begin* (c) *scfsi* (d) *part2_3_lenght* (e) *table_select* (f) *main data*

5.3 Security analysis

To this point, we can consider the main_data_begin as the most suitable part for the selective encryption of MP3 bit-streams. The security of the proposed encryption method is next analyzed. Obviously, there are two possible approaches one can use to attack the encryption scheme, that is, by attacking against the encryption algorithm or against the cipher text. Let's consider the first approach. It is reported in [7, 8, 11] that the algorithm 'RC4' used in our proposed scheme has been widely accepted to provide an adequate security

level for a secure data transmission. An efficient attack against RC4 is thus presumed presently unavailable. However, in the future, when this algorithm is found optimized; a different encryption algorithm that provides a stronger security level can be used instead. For the second approach, the easiest way to attack against the ciphertext is by trying to decode the encrypted bit streams using all possible patterns of the important parts. This technique is widely known as an enhancing search or brute force attack. In the proposed scheme, the output generated from the RC4 algorithm consists of a string of 8-bit numbers, and is directly XORed with the *main_data_begin*. Since each frame contains only one *main_data_begin*, one needs to try for 256 possible patterns to decode an encrypted frame. This concept can be applied to scfsi, *part2_3_length* and t*able_select* as well. Nevertheless, the latter two are harder to break since there will be 2^{16} possible patterns to try. Attacking the cipher text straightforward is hence equivalent to breaking the RC4 algorithm. Furthermore, this task becomes exponentially difficult when the attacker needs to break the entire bit-stream at once. For instance, in the worst case, the test file 'River' containing 1,531 frames will require 8×1,531 = 12,248 bits to be encrypted, and there will be up to $2^{12,248}$ possible patterns for the attacker to brake. The worst case here is understood in a way that all patterns of the *main_data_begin* within the test file 'River' are totally different.

To break the encryption scheme in a more efficient way, one may try to guess the pattern of the *main_data_begin*. However, referred to the MP3 coding standard and the bits reversed technique [12], this pattern may be varied each time the audio is encoded, depending on several factors e.g. characteristics of audio, encoding mode, hardware. Hence, if the attacks can break the first encrypted frame, they can sometimes guess the pattern of the next encoded frames. Normally, there are three possible patterns occurring within the successive MP3 frames, i.e. both are the same, slightly different or totally different. Although this attack is possible, the attacks must listen to the decoded audio every time to determine whether such improperly decoded MP3 bit-stream is intelligible or not. When the attackers need to break the entire MP3 bit-stream, this task becomes extremely difficult and requires a very large amount of time. That is, in the worst case of 'River', the attacker must listen to the played back decrypted MP3 file up to $39.84\times2^{12,248}$ seconds.

6. Conclusions

An implementation of selective encryption for MP3 compressed audio has been presented in this paper. Concisely, an important part of the MP3 bit-streams was encrypted, and that encrypted part was then used to protect the contents of the remaining unencrypted audio. The experimental results have shown that the *main_data_begin* extracted from the side information part of the MP3 bit-streams was the most suitable part used for selective encryption. The proposed method achieves both lower complexity and equivalent security level compared to the ordinary encryption method. On average, the SNR, the listening test score, the encryption percentage and the processing time measured from the proposed and ordinary encryption schemes were -3.56 dB, 4.84, 0.39 %, 21.50 ms and -4.24 dB, 4.88, 91.89 %, 7.57 s, respectively. In summary, our approach provides a low cost encryption for the MP3 compressed audio.

7. Acknowledgment

The authors would like to thank the reviewers for their valuable suggestions that greatly help improve the quality of this manuscript.

8. References

1. Amornraksa, T., 2001, *Data Security for Multimedia Communication*, King Mongkut's Institute of Technology Thonburi, pp. 56-72.

2. Thorwirth, N.J., Horvatic, P., Weis, R., and Jian Z., 2000, "Security Method for MP3 Music Delivery", *Proceedings of the 34th Asilomar Conference on Signals, Systems and Computers 2000*, Vol. 2, Oct. 29 - Nov. 1, pp. 1831-1835.

3. Torrubia, A. and Mora, F., 2002, "Perceptual Cryptography on MPEG 1 Layer III Bit-Streams", *Proceedings of International Conference on Consumer Electronics (ICCE 2002)*, June 18-20, pp. 324 - 325.

4. Gang, L., Akansu, A. N., Ramkumar, M., and Xuefei, X., 2001, "On-Line Music Protection and MP3 Compression", *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, May 2-4, pp. 13 - 16.

5. Servetti, A., Testa, C., and Martin, J. C., 2003, "Frequency-selective Partial Encryption of Compressed Audio", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, Vol. 5, April 6-10, pp. 68-71.

6. MPEG 1 Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mb/s, 1993, ISO/IEC 11172.

7. Tsoi, K.H., Lee, K.H., and Leong, P.H.W., 2002, "A Massively Parallel RC4 Key Search Engine", *Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'02)*, California, USA, April 21-24, pp. 13-21.

8. Jovan, D. G., 2000, "Iterative Probabilistic Cryptanalysis of RC4 Keystream Generator", *Proceedings of 5th Australasian Conference on Information Security and Privacy (ACISP 2000)*, Brisbane, Australia, July 10-12, pp. 220-233.

9. SQAM, 2002, Sound Quality Assessment Material Recoding for Subjective Tests [Online], Available: http://sound.media.mit.edu/mpeg4/audio/sqam, [10-24-2004].

10. MPEG Organization, 2002, Test Patterns for ISO/MPEG 1 Layer III [Online], Available: http://mpgedit.org/mpgedit/testdata/mpegdata.html#ISO_m2l123, [10-24-2004].

11. Burnett, S. and Paine, S., 2001, RSA Security's Official Guide to Cryptography, 1st ed., McGraw-Hill Osborne Media, pp. 124-136.

12. Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 3: Audio, 1993, 1st ed., ISO/IEC International Standard IS 11172-3.