New Encryption Approaches to MP3 Compression
Chih-Hsu Yen, Hung-Yu Wei, and Bing-Fei Wu
Department of Electrical and Control Engineering
National Chiao Tung University
1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C..
Taiwan R.O.C.

Abstract: Three new partial/adaptive encryption approaches to secure MP3 compression algorithm are presented. To adopt the proposed approaches, a cryptosystem can be chosen to be embedded into or be concatenated behind the MP3 algorithm, depending on the different characteristics of applications. The MP3 is a popular format for audio distribution, hence securing MP3 will provide diversified applications, such as music trial services, authorized access, and multilevel encryption. To encipher a compressed file by partial encryption, the encrypted bits will be diffused after decompression, hence the partial/adaptive encryptions are suitable for multimedia security. In this paper, the audio data are separately enciphered by these proposed approaches: sign bits of frequency magnitudes, Huffman codes and side information, and the results are analyzed in Masking to Noise Ratio sense. The proposed secure MP3 algorithm can be easily achieved without extensive computation, major modifications for MP3, and loss of the compression ratio. Moreover, the partial encryption can exactly provide enough security on multimedia application. And the encrypted MP3 files is compatible to MP3 standard for trial service. In this paper, we also give an example on how to build a trial version of MP3 files by our approaches.

Key-Words: Multimedia, Security, MP3 music, Encryption, Audio

1 Introduction
The digitalization of media is a profound impact on copyright and intellectual property. The past phenomenons reveal that the online MP3 sharing seems to threaten the music industry. Variety propositions on this issue are addressed by professionals and industries, and one of them is to well employ MP3 as one part of business models, hence the modifications or adoption ways on MP3 have to be considered. Therefore, it’s important to develop multimedia security technology. Thorwirth and Horvatic [9] encrypted audio data with selected frequency and provided several audio qualities by selective encryptions. Torrubia and Mora[8] proposed an idea of perceptual cryptography which encrypts data with an error factor ρ which determines the audio quality. Also, in [8], the random sequence of PRNG is used to substitute the codewords in Huffman tables and XORs with scale factors. Gang et al[1] concluded that MP3 files can be encrypted in compression progress or after compression.

Usually, multimedia encryption is just to cascade a cryptosystem behind a source encoder. Indeed, the multimedia encryption has been especially designed for practical applications. Source coding is data-dependent and compacts the data size; however, secrecy coding is data-independent and keeps the data size. Because of the different natures between source coding and secrecy coding, the source-secrecy coding is hard to design. At general points, designs of cryptosystems do not consider the properties of data. In fact, the characteristics of the the multimedia content have to be taken more consideration to achieve more efficiency on processing and more flexibility on applications while applying a cryptosystem on a compression algorithm. The diffusion, which is an important feature for substitution-permutation ciphers, also appears on decompression, hence the encrypted bits will extensively infect the decompressed multimedia content, i.e., the effects on quality of partial encryption may be as well as of full encryption. Additionally, the partial encryption can process less data and is suitable in real-time applications.

Our schemes can be implemented into two cases: simultaneously encrypting and compressing or encrypting the already encoded MP3 files. For live applications, which the former case is fit, the media
streams are encoded then are delivered immediately. Since it is a time critical issue, so it would be better to do encryption and compression simultaneously. For other applications, such as MP3 providers, because the compression is time-wasting processing, it would be fine enciphering compressed files than doing encryption in compression step. In this paper, we propose several approaches to secure MP3 to meet the above cases. The first approach is the sign-bit encryption: the sign bits of frequency magnitudes are treated as plaintext. The second method encrypts the Huffman codes of quantized frequency magnitudes. Both above schemes are applicable to adaptive encryption. In adaptive encryption, the information must be recorded in headers for receivers, hence we devise a way of recording the extra information to be compatible the MP3 standard. The last approach enciphers the side information in MP3 headers.

Section 2 gives brief descriptions on Masking to Noise Ratio (MNR), MP3, and Software encryption algorithm (SEAL) and the definitions of partial encryption and adaptive encryption. The three proposed schemes and their performance are dilated on in Section 3. The simulation results are shown and explained in Section 4. The stream cipher SEAL is chosen in our schemes to perform encryption, but not limit which ciphers to be adopted for our approaches. In Section 5, the realization issues, security of three algorithms, and solutions for trial service are described here. Finally, Section 6 summarizes the paper and provides direction for future work.

2 Preliminaries

2.1 Partial Encryption and Adaptive Encryption

The partial/adaptive encryption is a particular derivative from the combination of a cryptosystem and a source coding or an error control coding. The main concept of the partial/adaptive encryption is to protect the entire content by only encrypting the significant part which has smaller size. Generally, the encryption is behind compression. Hence, as shown in Fig. 1, the original signal $M$ is processed by filters or compression algorithm to obtain $Y$. The significant part can be determined among the generation of $Y$, and we denote it as $Y_c$, the part to be encrypted. Therefore, in the partial/adaptive encryption, $Y$ is divided into an encrypted part $Y_e$ and a clear part $Y_c$. The transmitted signal $R$ is the combination of $Y_c$ and $E(k, Y_e)$. Signal $\hat{M}$ is the reconstruction of $R$ without decryption, and we can compare $M$ and $\hat{M}$ to examine how the influence $E(k, Y_e)$ has. A scheme which makes some cumbrance on sensible presentation and causes that the polluting size of $M$ is larger than the size of $Y_e$ is called partial/adaptive encryption, as shown in Fig. 2.

![Fig. 1. The model description of partial/adaptive encryption](image1)

Both schemes are similar in the concept of encryption, but not in the applications. Hence we give definitions of partial encryption and adaptive encryption below.

![Fig. 2. The data flow of partial/adaptive encryption](image2)

**Definition 1:** Partial encryption is an encryption method which brings about the larger size of infected part of $\hat{M}$ than the size of $Y_e$ and some cumbrance while displaying $\hat{M}$, but the quality of $\hat{M}$ is too troublesome to be determined by the size of $Y_e$.

**Definition 2:** Adaptive encryption is an encryption method which brings about the larger size of infected part of $\hat{M}$ than the size of $Y_e$ and some cumbrance while displaying $\hat{M}$, but the quality of $\hat{M}$ can be determined by the size of $Y_e$ in a systematic way.

Both schemes have a common property that the complete reconstruction can’t be gained without decryption or authorized access. The partial/adaptive encryption relies upon how to select $Y_e$. The only principle for $Y_e$ selection is to pick the $M$-dependent data to encrypt. There are two types of the $M$-dependent data.
1) Variant of $M$: This kind of data can be truly regarded as signal $M$ of other formats, in other words, it is just some transformation of $M$, for examples, the results of DCT (Discrete Cosine Transform), or Huffman coding of $M$.

2) Accompaniment of $M$: The input-dependent and additional information, which receivers needed to decode, are belong to this type, for examples, Huffman table, CRC code, and frame headers.

Generally, the partial encryption encrypts the later type and the adaptive encryption takes the other.

The MP3 algorithm has both pre-described types of $M$-dependent data; actually, most applicable algorithms do. Hence our proposed schemes include both partial encryption and adaptive encryption.

### 2.2 Masking-to-noise ratio

We use MNR to analyze the audio quality after MP3 compression and encryption. MNR can be obtained from the masking model of human hearing, so it’s better than SNR to represent the audio quality. The MNR is defined in (1).

$$MNR = 10 \log_{10} \left( \frac{\text{power_of_masking_threshold}}{\text{power_of_noise}} \right). \tag{1}$$

![Fig. 3. The general model of a lossy compression](image)

Fig. 3 shows the general diagram of a lossy codec. An original signal $X[n]$ is compressed as $Y[n]$, then the error between $X[n]$ and $\hat{X}[n]$ is defined as

$$E[n] = X[n] - \hat{X}[n].$$

To have MNR, the masking threshold is calculated from the results of feeding psychoacoustic model with $X[n]$ and the noise energy is computed by $E[n]$. When MNR is greater than zero, human hearing is hard to detect the noises; more exactly, noises are masked. For a compressed audio, each frequency band has its own MNR. For convenient comparison, we use the mean value of MNRs to represent the audio quality.

### 2.3 MP3 algorithm

The MP3 encoding algorithm is shown in Fig. 4[6]-[4]. The original audio signals are analyzed by a 32-channel polyphase analysis filterbank. The modified DCT(MDCT) transfers the segmented-by-frequency data into frequency magnitudes, then each magnitude is quantized, then is compressed by Huffman coding. In MDCT, the short window is applied before DCT to decrease the echo. The original data are also analyzed by FFT to provide the information needed by psychoacoustic model II which gives the signal-to-mask ratio SMR value to switch window length and bit allocation. At the last stage, all frames are appended with MP3 headers. Fig. 5 shows the file format of MP3. In the header field, it contains the information of sampling frequency, bit rates, audio modes, etc.; the CRC is adopted to detect whether errors occurred on fields of header and side information. All related parameters of decoding information are in the side-information field. Finally, the field of main data contains the compressed results of audio data.

![Fig. 4. The block diagram of MP3 encoding algorithm.](image)

![Fig. 5. The file format of MP3[4].](image)

### 2.4 SEAL

The SEAL[7] is a software optimized stream cipher with key length of 160 bits. For the case of the processors with eight registers, SEAL can keep the number of involving variables less than 8. Most operations in SEAL only have two operators and the table size is less than 4K bytes, which is especially designed for the processors with small on-chip cache. SEAL uses secure hash algorithm(SHA)[5] to generate three tables, which are needed by initialization and key generation. According to Table 1, SEAL is more efficient than most stream ciphers and that’s why we apply it in our work.
3 Our Proposed Schemes

Based on the designed principle and definitions of partial encryption and adaptive encryption, we proposed three approaches, sign-bit encryption, Huffman-code encryption, and side-information encryption, to secure MP3 algorithm. All proposed schemes can be applied by embedding a cipher into MP3 encoder or using a cipher to straight encrypt the MP3 files.

3.1 Sign-bit encryption

In MP3, the frequency magnitude for each sample has two parts: an absolute value and a sign bit. When the value of the sample is less than 0, the sign bit is set to 1; otherwise, it is set to 0. The human hearing is sensitive to the variation on sound stemming from changing sign bits of samples below 5 kHz[2][3], hence the audio quality is dropped significantly while encrypting those sign bits. Encrypting the sign bits above 5 kHz is not easy to be observed by audiences, though the energy is twisted. According to the properties discussed above on sign bits, that encrypting sign bits could achieve the adaptive encryption is obvious.

For each granule in MP3, there are 576 samples with equal bandwidth, and we denote the set of sign bits of $i^{th}$ granule as $S_i = \{s_{i,j}\}$ the sign bit of the sample $j$, $0 \leq j \leq 575$, at the $i^{th}$ granule. The encrypted part of $i^{th}$ granule, $Y_{e,i} = \{s_{i,k} | s_{i,k} \in S_i$ and $a \leq k \leq b$, where $[a, b]$ is the interval of granule to be encrypted}. If the full-band encryption is selected, we encrypt the entire set $S_i$, i.e., the special case of $a = 0, b = 575$; otherwise, we can adjust the music quality by selecting which bands to encrypt, while the encrypted MP3 files are directly played by original MP3 player.

Fig. 6 shows the MNR distribution in several cases. In Fig. 6(b), MNRs are greater than zero for all bands, but drop slightly at frequency about 5 kHz, as compared with Fig. 6(a). But Fig. 6(c) reveals the poor performance in MNRs within the frequency range from 0 to 5 kHz, even having some MNRs smaller than zero. As opposed to Fig. 6(a), that decrements of MNR in Fig. 6(c) is more than the ones in Fig. 6(b) reveals the fact that human hearing is sensitive to the variation stemming from changing sign bits of samples below 5 kHz[2][3].

3.2 Huffman-code encryption

In MP3 definition, each frame has four granules, each granule has 576 MDCT coefficients. The absolute values of these 576 coefficients will be divided into three regions: big value, count one region, and zero region, and be sequently compressed with four Huffman tables. We denote the Huffman code $W_i = \{w_{i,0}, w_{i,1}, \ldots, w_{i,575}\}$ as the whole Huffman codeword of 576 coefficients at the $i^{th}$ granule, where $w_{i,j}$ is the Huffman codeword of the $j^{th}$ quantized coefficient.

The Huffman code $W_i$ is made of concatenating each codeword of samples sorted by frequency from low to high. However, when choosing some bits of $W_i$, that distinctly pointing out their corresponding frequency is hard. We can only know the information that the corresponding frequency of $w_{i,j}$ is lower than the one of $w_{i,k}$, when $j < k$.

Similarly to sign-bit encryption, the quality of unauthorized accessing can be adjusted by selecting the bytes of $W$ to encrypt, so Huffman-code encryption is an adaptive encryption. Different from sign-bit

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mbit/s</th>
<th>Relative speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEAL</td>
<td>198</td>
<td>1</td>
</tr>
<tr>
<td>RC4</td>
<td>110</td>
<td>1.8</td>
</tr>
<tr>
<td>RC5-32/12</td>
<td>38.4</td>
<td>5.2</td>
</tr>
<tr>
<td>DES</td>
<td>16.9</td>
<td>11.7</td>
</tr>
<tr>
<td>MD5</td>
<td>133.1</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Fig. 6. The simulation of sign-bit encryption on different frequencies: (a) The original MP3 file; (b) The sign bits above 5kHZ are encrypted; (c) The sign bits below 5kHz are encrypted.

TABLE 1
Performance of several ciphers on Intel Pentium processor[7]
encryption, an error avalanche occurs in Huffman-code encryption, because Huffman coding has an error propagation problem, even encrypting the first byte will cause tremendous errors in decompression.

The $W_i$ is a variant of 576 MDCT coefficients, hence the encrypted part $Y_{e,i}$ can be determined from $W_i$ as the plaintext in Huffman-code encryption. Because the bit length of $w_{i,j}$ is variable, not multiple of byte, it is inconvenient to pick a set $Y_{e,i} = \{w_{i,j}|w_{i,j} \in W_i\}$, and $a \leq j \leq b$, where $[a \ b]$ is the interval to be encrypted. Therefore, we select $Y_{e,i}$ in byte.

### 3.3 Side-information encryption

Side information records the data needed by MP3 encoder. There are several fields of side information. Some are media dependent and some are not. For security concern, we have to avoid known-plaintext attacks, hence much attention have to be paid on encrypting header-like data.

In our scheme, we choose those fields which changed by input media as plaintext. In MP3, all fields are classified into three kinds: frame side information, channel side information, and granule side information. Literally, frame side information is for each frame, and so as channel side information and granule side information. We choose some side information, of which symbols are defined in MP3 specification [4], to keep the side-information encryption from the known-plaintext attacks and list them below.

1) **Frame side information**: main_data_bigin indicates the beginning of compressed data, hence an MP3 encoder won’t correctly decode without possessing this information.

2) **Granule side information**: These fields are the decoding information, such as quantizer step, region boundary, Huffman table, and so on.
   a) part2_3_length: this value contains the number of main data bits used for scale-factor and Huffman code data.
   b) big_values: the spectral values of each granule are coded with different Huffman code.
   c) global_gain: the quantizer step size information is transmitted in the side information variable global_gain.
   d) scalefac_compress: selects the number of bits used for the transmission of the scalefactors.
   e) region_address: a further partitioning of the spectrum is used to enhance the performance of the Huffman coder.

f) **table select**: different Huffman code tables are used depending on the maximum quantized value and the local statistics of the signal.

Additionally, the CRC is encrypted necessarily to probably keep away from attacks, because CRC implicitly manifests some relationship among all fields in side information. All the listed fields are accompaniments of input signal, so side-information encryption is a partial encryption. It is obvious that the quality can not be systematically adjust by choosing the side information.

### 4 Simulation Results

![Fig. 7. The simulation of sign-bit encryption on different frequencies in three types of music, piano, rock, and pop. The horizontal lines are the MNR of original MP3 files.](image)

We use SEAL to encrypt and simulate the proposed schemes with three types of music: piano, rock, and pop. We discuss the results over the average MNR. The frequency distribution of piano, rock, and pop are $0 \sim 8$ kHz, $0 \sim 22$ kHz, and $0 \sim 16$ kHz, respectively.

In **sign-bit encryption**, we simulate with encrypting different bands. As shown in Fig. 7, when the encrypted frequency is increased, the audio quality is also improved. As we mentioned above, upon 5 kHz, the variation of sign bits will not vastly drop the
quality. This simulation also tells one thing that **sign-bit encryption has tolerable quality with encrypting a section between 4 ∼ 6 kHz for music trial service.**

![Fig. 8](image)

**Fig. 8.** These results are obtained by encrypting Huffman codes. The x-axis is the number of encrypted bytes counted from the first byte of Huffman codes.

In *Huffman-code encryption*, because we can not exactly have the frequency information on Huffman codes, we use byte index instead of frequency index as x-axis. In spite of what kinds of music, from Fig. 8, we found no matter one byte or several bytes are encrypted, the quality of encrypted MP3 file are similar under *MNR* sense. The effects are resulted from the property of error propagation on Huffman decoding.

Different from the simulation of Fig. 8, instead of encrypting consecutive bytes, we separately encrypt each byte. The simulation results provide us that encrypting which byte is proper to music trial service. However, the *Huffman-code encryption* is hard to determine which frequency band is good for music trial service. The byte index only shows the relative information about frequency. For examples, even the 2nd byte implies higher frequency than the 1st byte in each granule, but the 1st bytes of distinct granules do not represent the same frequency. The Huffman codes are the compressed data of nonzero region. For piano music, most of energy are concentrated in low bands and wide bandwidth in zero region, hence the quality of piano increases slowly while the frequency of encrypted band increases.

The results of side information encryption are illustrated in Table 2. From the *MNR* values, the MP3 files have been successfully encrypted. The important information are mostly in frame side information and granule side information, hence encryption on both of them provide good multimedia security.

![Fig. 9](image)

**Fig. 9.** These results are obtained by encrypting Huffman codes. The x-axis is the byte index of the encrypted byte.

<table>
<thead>
<tr>
<th>Non</th>
<th>Frame</th>
<th>Channel</th>
<th>Granule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piano</td>
<td>29.3831dB</td>
<td>20.6328dB</td>
<td>23.6214dB</td>
</tr>
</tbody>
</table>

**TABLE 2**

| MNR COMPARISON: SIDE INFORMATION ENCRYPTION ON PIANO, ROCK, AND POP. |

For high security application, the **sign-bit encryption** encrypts the sign bits of samples below 5 kHz, the **Huffman-code encryption** takes 1st ∼ 70th bytes of each granule as Y_e,i, and **side-information encryption** enciphers all side information. Table 3 lists the encrypted size for each case.

![Table 3](image)

**TABLE 3**

| SIZE OF Y_e FOR HIGH SECURITY OF THREE SCHEMES |

<table>
<thead>
<tr>
<th>Sign-bit encryption</th>
<th>Huffman-code encryption</th>
<th>Side-information encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{T} ) (%)</td>
<td>15.56% (&lt; 5 kHz)</td>
<td>19% (0 ~ 7.65% (256 bits)</td>
</tr>
<tr>
<td>20 bytes)</td>
<td>20 bytes)</td>
<td>20 bytes)</td>
</tr>
</tbody>
</table>

\( T = 20 \) bytes
5 Implementations and Security

Based on the simulation results, some practical issues and suggestions are addressed in following subsections.

5.1 Quality Level for Trial Services

According to the simulation results in Section 4, we suggest a quality level for music trial services for the both proposed adaptive encryption, sign-bit encryption and Huffman-code encryption, to original MP3 players. In sign-bit encryption, encryption of bands about 1, 225 ∼ 3, 062 Hz could obtain the music for trial service. For Huffman-code encryption, the quality for trial services could be provided by encrypting the 60th ∼ 90th bytes of Huffman codeword in each granule. The results are shown in Table 4.

<table>
<thead>
<tr>
<th>Sign-bit encryption</th>
<th>Huffman-code encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 225 ∼ 3, 062 Hz</td>
<td>60th ∼ 90th bytes</td>
</tr>
<tr>
<td>MNR</td>
<td>22 db</td>
</tr>
<tr>
<td></td>
<td>23 db</td>
</tr>
</tbody>
</table>

If the audio quality is not highly concerned, the devised three schemes can fully encrypt the corresponding encrypted parts $Y_e$: sign bits, Huffman codes, and side information. If there exits the time issue of processing, the data size of $Y_e$ can be reduced by selecting those parts $Y_e$ from the low frequency bands to the high bands for sign-bit encryption or from low bytes to high bytes for Huffman-code encryption.

5.2 Extra Header Design

Taking inspection on side-information encryption firstly, because the quality adjustment is unsuitable, for realization, it is a good strategy that the encrypted side information is beforehand defined and known by encoders and decoders. So there doesn’t need extra information to indicate how to decrypt with full quality, and standard MP3 players could decode the side-information encryption MP3 files as usual ones, but with low quality.

For sign-bit and Huffman-code encryptions, while the quality adjustment is available in encryption, saving the information of how many data are encrypted is necessary. In order to be compatible with the standard MP3 players, the information, needed for correct decryption, has to be added into standard MP3 header in a proper way so that the standard MP3 players can play the encrypted MP3 with low quality.

On sign-bit and Huffman-code encryptions, we can place the extra information prior to the synchronization word, because the MP3 decoder takes no consideration on the input before detecting the synchronization word, i.e., standard MP3 players can play the encrypted MP3 files as well. Without doubt, the music of full quality can be gotten under success authorization and specific MP3 decoders.

The synchronization word of hex is 0xFFF, hence the construction of extra information must have no probabilities to produce the pattern, 0xFFF. In sign-bit encryption, the encrypted part $Y_{e,e}$ can be described by two 10-bit digits ranging from 0 to 575, therefore the cases with the maximum length of consecutive 1’s are \{0111111111000000\} in all possible combinations of the two digits. It is apparent that the pattern 0xFFF by no means occurs in extra information for sign-bit encryption. In Huffman-code encryption, we also use two 10-bit digits to define the bound of bytes to encrypt. Generally, the Huffman codeword size of a granule is less than 512 bytes, the most 1’s case is \{011111111101111111\}. For that reason, the extra information of Huffman-code encryption is certainly not to get the same pattern as synchronization word.

5.3 On-Line and Off-Line Encryptions

The proposed three schemes can on-line or off-line encrypt MP3 files according as the practical applications. In general, the on-line applications are time critical, such as live broadcasts of sport games, and concerts, so it’s better to produce encrypted MP3 files by simultaneously encrypting and compressing the raw data. And the off-line instances, therefore, can be done by directly encrypting the ready MP3 files.

The on-line cases are composed of MP3 algorithm and SEAL, each case has different insertion position of SEAL as shown in Fig. 10. The off-line cases have to extract the encrypted part $Y_e$ from the ready MP3 files, then encrypt $Y_e$. The encrypted part $Y_e$ of side-information and Huffman-code encryptions are directly extracted from each frame of a MP3 file. In off-line sign-bit encryption, however, the sign bits to encrypt are obtained after Huffman decoding of each frame. The pictorial description is at Fig. 11.

5.4 Security

Apparently, the proposed schemes are cipher independent, hence the security on our schemes are dependent
on what kind of data we encrypted. For this concern, we have noted whether the encrypted data are predictable or known already by cryptanalysts. For sign-bit encryption, cryptanalysts are not easy to get or predict the sign bit of each sample. In Huffman-code encryption, the compressed data are the variant of the original signal, hence it has similarly intrinsic property as sign-bit encryption, that is, the compression results are a huge alphabet to guess. However, some fields of side information are still and simple to predict. For security concern, we have to avoid encrypting those fields and keep our side-information encryption from known-plaintext attacks.

6 Conclusion

We propose several approaches to secure MP3 in this paper. They provide an easy solution to integrate the cryptosystem and MP3 algorithm. If the adaptive encryption is needed, the sign-bit encryption and Huffman-code encryption can fit in with, along with good security. When the size of encrypted data are fixed, the side-information encryption provides good security than the others. Our schemes can be applied on simultaneously encrypting and compressing or encrypting the already encoded MP3 files. For live applications, e.g., diverse sport shows, and concerts, the former case is fit, because the media streams are encoded then are delivered immediately; otherwise, because the compression is time-wasting processing, it’s better enciphering compressed files than doing encryption in compression step. The partial/adaptive encryption and the selection principle of $Y$, addressed in Section 2 are applicable not only on MP3 algorithm but also other compression algorithms.

Additionally, simulation results are given as criteria of the decision of quality-security or security-time dilemma.

Acknowledgement

The work was supported by the Program for Promoting Academic Excellence of Universities under Grant No. 91X104-EX-91-E-FA06-4-4.

References