SRMT: A Lightweight Encryption Scheme for Secure Real-time Multimedia Transmission [*]

Euijin Choo, Jehyun Lee, Heejo Lee
Korea University
{chksj,arondit,heejo}@korea.ac.kr

Giwon Nam
Xenolink Communications
gwnam@xenolink.co.kr

Abstract

Securing multimedia transmission has become a challenging issue due to the popularization of real-time multimedia applications such as video surveillance, satellite communication and web cams. However, previous security algorithms do not always guarantee a satisfactory degree of media quality and latency. In order to provide both security and media QoS, a viable security mechanism must consider three properties: processing time, compression rate and security level. In this paper, we propose a light-weight encryption scheme without loss of security and media QoS, called Secure Real-time Media Transmission *(SRMT) using two block transpositions and a XOR operation. SRMT is studied with respect to MPEG-4, which is widely used in today's multimedia applications. Experimental results with various MPEG-4 movies show that the SRMT scheme achieves real-time transmission of encrypted media data without loss of security and media QoS. Though SRMT is conducted on uncompressed raw data, SRMT encrypts 3 times faster than the AES encryption of MPEG compressed data. Also, we show that manipulating key frames and a compression method can lessen increasing ratio of encrypted MPEG size, e.g., 70.5% improvement over an existing combination method of block transpositions and XOR operations.*

1 Introduction

The rapid growth of multimedia and networking technologies gives rise to numerous multimedia applications, such as video surveillance, satellite communication and web cams [15]. Consequently, securing multimedia transmission has become a challenging issue. Due to the unique characteristics of real-time multimedia data such as large data size, high bandwidth and real-time requirements [12, 15], a proper security algorithm should be chosen carefully for real-time multimedia transmission. This is due to the fact that media QoS should be met to provide high media quality and low latency even when securing media transmission. In order to preserve high media QoS, a viable security mechanism should provide three properties: high-speed processing, high compression rate and sufficient security level. However, previous cryptographic approaches have considered only one or two of these properties. For instance, security has been diminished in order to improve processing speed [1, 4, 7]. Enhancing security has resulted in the drop of frames sacrificing media QoS for real-time requirements [9, 10].

In this paper, we propose a lightweight encryption scheme without loss of security and media QoS, called *Secure Real-time Media Transmission*(SRMT). Since MPEG-4 is widely used in modern media transmission, SRMT is studied with respect to MPEG-4. We implemented the SRMT scheme along with other encryption mechanisms. Experimental results with various MPEG-4 movies show that we achieve real-time transmission of encrypted media data without loss of security and media QoS.

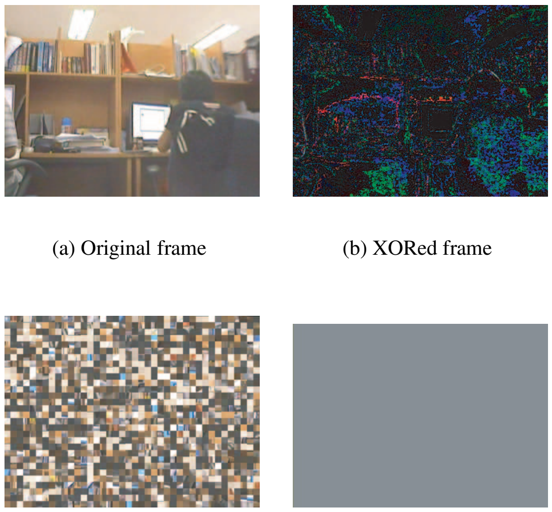
The remaining of this paper is organized as follows. Section 2 presents related works. In Section 3, we propose a lightweight encryption scheme, SRMT. Section 4 discusses the experimental results of SRMT and analyze the SRMT. Finally, Section 5 concludes the paper.

2 Related Work

For securing media transmission, many encryption algorithms have been proposed with objectives classified into two categories: security and light-weightness.

Among light-weight encryption mechanisms, encryption with XOR has been considered as the simplest encryption mechanism [6, 8]. However, the XOR encryption produces the inverse color image which has an original outline as shown in Fig. 1(b). It implies that XOR encryption sacrifices security for light-weightness. In order to enhance

[*]This work was supported in part by the ITRC program of the Korea Ministry of Information & Communications, and the Basic Research Program of the Korea Science & Engineering Foundation



(a) Original frame (b) XORed frame

(c) Scrambled frame (d) AES encrypted frame

Figure 1. Three existing algorithms for encrypting video images.

security, XOR encryption can be used together with another encryption algorithm such as DES or AES [7, 15]. However, they are too slow and heavy for handling large multimedia data [3, 8, 11, 14]. XOR-only encryption is lightweight and fast enough, but it diminishes security as well as compression rate significantly.

Several stream ciphers like RC4 is also lightweight encryption algorithm. If RC4 encryption which randomizes data [5] is applied for uncompressed multimedia data, it has resulted in remarkably low compression rate of encrypted MPEG. Therefore, RC4 encryption can be employed for compressed data [4]. Even if RC4 encryption is used for compressed data, processing speed of RC4 encryption is inadequate for real-time multimedia transmission [4]. Also, the weakness of RC4 encryption such as key scheduling algorithm and state table of RC4 was shown in many researches [5].

One approach to increase the processing speed is encrypting chosen parts of images selectively, which reduces the amount of data to be encrypted. There have been several selective encryption methods for light-weightness, but these methods do not provide sufficient security [1, 4, 7].

Block scrambling [11, 14] and the MVEA(Motion Vector Encryption Algorithm) [3] are media adaptive encryption methods. Block scrambling appears secure as shown in Fig. 1(c), and provides high compression rate. However, it was shown that scramble-only methods do not provide sufficient security [2, 14]. MVEA has been considered as secure and light-weight [3]. However, MVEA manages only P frames, and we need additional encryption for I frames [3]. In [3], for securing I frames, MVEA was used with DES which incurs the severe loss of light-weightness.

AES can be considered as a secure encryption method as shown in Fig. 1(d). However, AES is inadequate for media data since it is too heavy and slow for handling large media data [3, 8, 11, 14]. Furthermore, AES provides very low compression rate.

3 The SRMT Encryption Scheme

This section describes the proposed encryption scheme called *Secure Real-time Media Transmission*(SRMT). For the purpose of light-weightness, several selective encryption algorithms have been proposed. But selective algorithms provide lower security than the naive algorithm that encrypts all data. Since we need both light-weightness and security, we combined two lightweight mechanisms, XOR and transposition, in an efficient way. Pseudocode in Fig. 2 describes SRMT encryption operations in detail. The SRMT scheme uses two block transpositions and one XOR operation. First transposition is for generating a key frame. A XOR operation and second transposition is main encryption process. In Section 3.1, we will describe the mechanism of key frame generation. And the main encryption process will be described in Section 3.2.

3.1 Key Frame Generation

A randomly generated key may greatly degrade compression rate. In order to preserve high compression rate of encrypted frame, we have devised a seed frame S. S consists of pixels whose color values are close to unusual color value in the target environment. Suppose that the color values between 0x64 and 0xC8 are commonly found and the color value 0xFF is rare in the target environment. Then, the color values close to 0xFF, e.g. the values between 0xFA and 0xFF, can be elements in the seed frame S. In order to improve compression rate, most bits in the key for the XOR operation must consist of either zero or one. If the percentage of zero bits is high in the key, the XORed frame would be almost the same as the original frame. If the key value is selected among color values commonly found in the target environment, it makes a great difference between the XORed values of similar values that are in the bounds of the key value. As a result, the compression rate of a frame after the XOR operation will be degraded significantly. The set of XOR keys for a frame, namely the key frame K, are generated by transposing the seed frame S. Sample key frames are shown in Fig. 3.

```
S = Seed frame
K = Key frame
O = The object frame for encryption
X = The XORed frame with O and K
E = Finally encrypted frame
key1[i] = Keys for first transposition
            (i = 1,...,block number of S)
key2[j] = Keys for second transposition
            (j = 1,...,block number of X)

Step 1. Transposition
for(i=0; i < block_num of S; i++)
begin
    Trans_pos [i] = key1[i] % block [i] of S
    K is generated.
end

Step 2. XOR
$X = O \oplus K,\ where \oplus is\ XOR\ operation$

Step 3. Transposition
for(j=0; j < block_num of X; j++)
begin
    Trans_pos [j] = key2[j] % block [j] of X
    E is generated.
end
```

Figure 2. The SRMT encryption algorithm using two block transpositions and one XOR operation

3.2 Main Encryption Process

After key frame K is generated, the main encryption part is processed. Fig. 4 represents the main encryption process. The main encryption process consists of a XOR operation and second transposition. Original frame O is XORed with K and then, the XORed frame X is transposed. Finally, sample encrypted frames are shown in Fig. 5. We combined selective and naive algorithms. The objects for the transpositions are all blocks, and the objects for XOR are all pixels. However, the color value of each pixel can be selectively XORed. For instance, only red and green color values of each pixel are XORed except blue color value. From the combination of selective and naive algorithms, we can achieve both security and light-weightness.

3.3 The SRMT Keys

To keep continuous security, key management is also an essential part of the system. In the SRMT scheme, we need three kinds of SRMT keys which take the form of the frame.

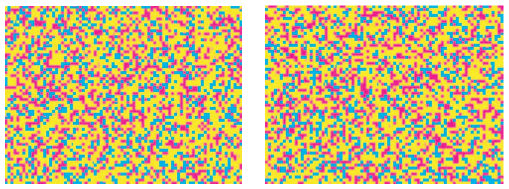


Figure 3. Sample key frames

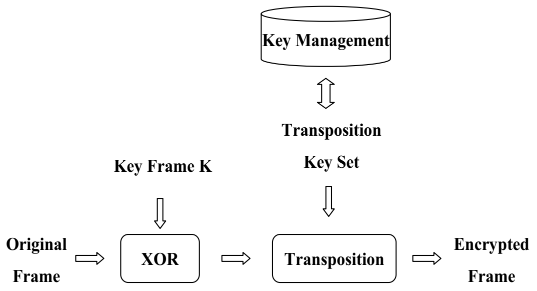


Figure 4. Main encryption process

One is the seed frame to generate the XOR key frame, and the others are transposition key sets(each block has a key) for two transpositions. Each block size for two transpositions can be changed periodically or optionally. Therefore, the key length corresponding to each block also changes depends on the alteration of transposition block size. The SRMT keys can be updated periodically for the purpose of sustained security.

4 Experimental Results and Analysis

In order to evaluate the performance of the proposed SRMT scheme, we implemented a network model as shown in Fig. 6. We used a DVR camera with an encoding / streaming server using MPEG-4. The hardware configurations of streaming server and clients for our experiment are Pentium4 3.0 GHz CPU and 1 GB of main memory with Windows XP SP2 as an operating system. To test our scheme, we modified the XviD video codec [13], which is publicly available and an ISO MPEG-4 compliant video codec. The media data is represented in resolutions of 640 $\times$ 480 pixels with a frame rate of 30 fps. We have already mentioned three essential properties: processing time, compression rate, and security level, for securing real-time media transmission with preserving high media QoS. In the following subsections, we analyze the SRMT scheme in terms of three properties.

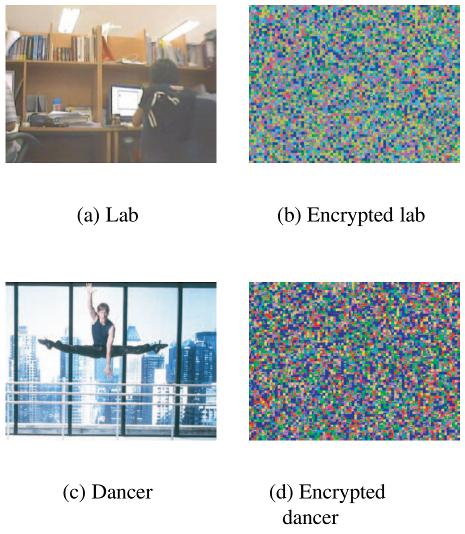


Figure 5. Encrypted frame examples

4.1 Security

We classified attack models into two categories: ciphertext only attack and chosen plaintext attack.

A. Ciphertext Only Attack: The worst situation for attackers is that they know neither plaintext nor the encryption algorithm. From the frame, attackers would be able to estimate that SRMT uses the XOR and transposition method, and transposition block size to some extent. In this case, to find out the transposition key, attackers would attempt an exhaustive attack using all possible transposition keys. Let L_t be the bit length of transposition keys for each block, N_b be the number of block, K_t be the set of transposition keys for all blocks of one frame, and N_{K_t} be the number of all $K_t's$. The complexity of restoring a frame is $2^{L_t * N_b}$. Given an encrypted frame only, suppose that attackers found a K_t. Then they could know transposition rules for a frame. Nonetheless, there is no assurance that the rule found by attackers is an accurate rule used by SRMT, since collision could have occurred as shown in Fig. 7. Fig. 7 represents that the same encrypted frame for a frame could be made by lots of different rules. The probability of reusing K_t is $1/N_{K_t}$. Since K_t's are updated periodically, it is practically impossible to exhaustively find out all key sets within a lifetime.

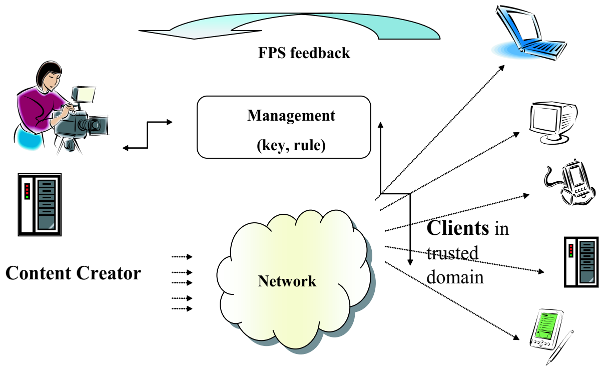


Figure 6. A network model for evaluating SRMT encryption scheme

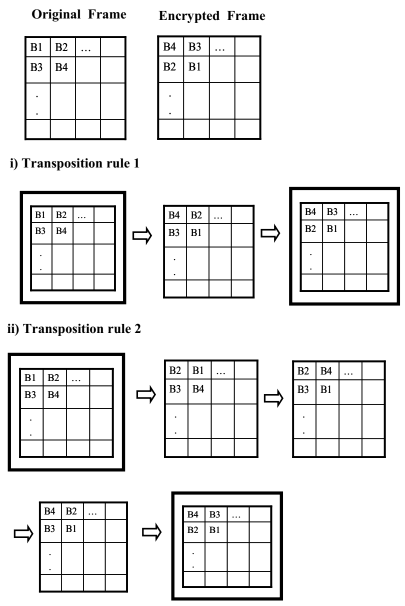


Figure 7. A collision example in the case that the attackers found transposition key sets for one frame

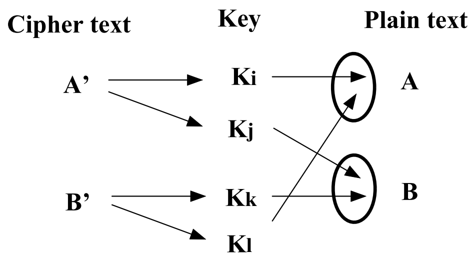


Figure 8. A collision example in the case that the attackers found a XOR key for a frame

To find out the XOR key, attackers would attempt an exhaustive attack using all possible XOR keys. Let w be frame width, h be frame height, K_X be XOR key, and N_{K_X} be the number of K_X's. The complexity of finding out K_X for one frame is 2^{w*h*24}. However, attackers would not be able to find out an accurate K_X, since collision could have occurred as shown in Fig. 8, even if the attackers have found a XOR key. Assume K_i and K_k are used keys for encrypting plaintext A and B to A' and B', respectively. However, A could be encrypted to B' with K_l. In this case, attackers have found any XOR key K_l, but it could never be used for encryption. In other words, even if attackers can restore any frame from time to time, there is very little probability that the attackers find the very key for which SRMT is used. And when attackers find an accurate XOR key K_X, the probability of reusing K_X is $1/N_{K_X}$. That is, the complexity for restoring one frame with key found by attackers is $2^{w*h*24} * N_{K_X}$. Since K_X's are updated periodically, there is little possibility to exhaustively find out all K_X's within a lifetime.

B. Chosen Plaintext Attack: The best situation that attackers can experience is when they have the knowledge of both plaintext and the particular SRMT encryption algorithm. Let C_S be the colors used in seed frame S, and N_{C_S} be the number of C_S. If attackers have no knowledge of C_S, attackers would be able to cryptanalyze the colors with the complexity of $2^{N_{C_S}}$. If attackers are aware of C_S, attackers could find out K_t by comparing the color gradations of each block for cipher frame and plain frame. The complexity of comparing the color gradations of two frames is as follows.

$$(N_b) + (N_b - 1) + (N_b - 2) + ... + 1 = \frac{N_b(N_b + 1)}{2}$$

In the case of finding K_t for a frame, attackers would be able to obtain the XOR key frame for the frame. By classifying the key frames, attackers would obtain the color composition of S. However, it is impossible for attackers to find out the arrangement of each C_S and K_t for a frame to make the XOR key frame. That is, it is impossible to reuse the XOR key frame for a frame found by attackers for decrypting another frame, except in the case that an identical XOR key frame is used in SRMT.

4.2 Compression Rate

Compression rate is an essential part of real-time media transmission, since the amount of media data to be delivered changes according to compression rate. In order to evaluate compression rate after encryption, we calculate encrypted MPEG size increasing ratio Φ of each algorithm over original MPEG size, as follows.

$$\Phi = \frac{(Encrypted\ MPEG\ size - Original\ MPEG\ size)}{Original\ MPEG\ size}$$

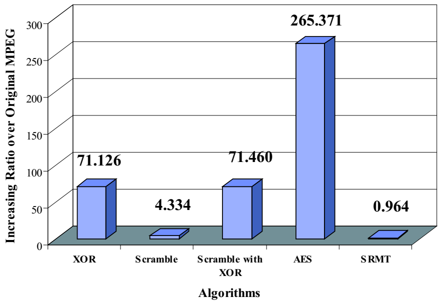
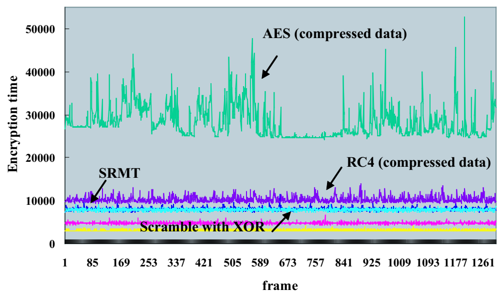


Figure 9. Encrypted MPEG size increasing ratio of each algorithm over original MPEG size

Fig. 9 represents Φ of existing encryption algorithms and SRMT encryption algorithm over original MPEG size. Most encryption algorithms have result in increase of MPEG size greatly as shown in Fig. 9. That is, MPEG compression rate after encryption is remarkably low in most case. Because P, B frames refer to the I frame when they are compressed in MPEG form. If P, B frames refer to the encrypted I frame, there is no similar I frame to the current P, B frame. That is, every frame is regarded as I frame, decreasing the compression rate of MPEG movies considerably. SRMT also uses a combination method of XOR and transposition, which is similar to existing combination methods. However, the compression rate is much higher than existing combination algorithms of block transpositions and XOR operations, because SRMT utilizes a specific key frame, and modified compression method. In SRMT, the P frame refers to the I frame encrypted by the same key as P frame separately, when P frame is MPEG compressed. Experimental results with various MPEG-4 movies show that manipulating key frames and a compression method can lessen increasing ratio of encrypted MPEG size, e.g., 70.5% improvement over an existing combination method of block transpositions and XOR operations which is shown in Fig. 9.

4.3 Processing Time

For secure real-time media transmission, we need to provide high-speed encryption. Fig. 10 represents the processing time of existing encryption algorithms and the SRMT. The SRMT is conducted on uncompressed raw data. Nevertheless, the SRMT mechanism encrypts 3 times faster than AES encryption for MPEG compressed data.



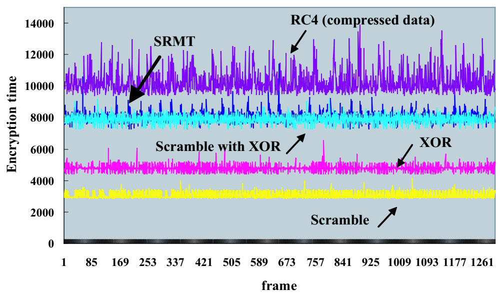


Figure 10. Encryption time(μs) of each algorithm

5 Conclusion

For the confidentiality of multimedia data, this paper has presented a light-weight encryption scheme without loss of security and media QoS, called *Secure Real-time Media Transmission*(SRMT) using a XOR operation and two block transpositions. In order to provide both security and media QoS, we consider three properties: processing time, compression rate, and security level. By using two light-weight mechanisms, XOR and transposition, the SRMT scheme achieve real-time transmission of encrypted media data without loss of security and media QoS. Experimental results with various MPEG-4 movies shows that SRMT scheme outperforms previous mechanisms in terms of media QoS and security. However, some challenges still must be surmounted. XviD codec used in this paper encodes the next block after the former block is saved. If SRMT transposed encoded blocks and not-encoded blocks, the encoder would not be able to find the position of the block that has not been encoded. That is, each block does not have its own motion vector information before encryption. Since B frame has only motion vector information, the MPEG compression rate of SRMT is sensitive to the quantity of B frames. To make the best use of motion vector information, that greatly contributes to the MPEG compression rate, it is necessary that every block has its own motion vector information before encryption. To achieve more improved MPEG compression rate, we have plans of modifying the XviD procedure sweepingly or carry out preprocessing to hold block information. And data integrity is also under consideration for more improved security level.

References

[1] B. Bhargava, C. Shi, and S. Wang. MPEG video encryption algorithms. *Multimedia Tools and Applications*, pages 57–79, 9 2004.

[2] B. Furht and D. Kirovski. *Multimedia Security Handbook*. CRC Press LLC, Dec. 2004.

[3] Z. Liu and X. Li. Motion vector encryption in multimedia streaming. In *Int. Conf. on Multimedia Modeling*. IEEE, 2004.

[4] Z. Liu, D. Peng, Y. Zheng, and J. Liu. Communication protection in IP-based video surveillance systems. In *Int. Symposium on Multimedia*. IEEE, Dec. 2005.

[5] S. Paul1 and B. Preneel1. A new weakness in the rc4 keystream generator and an approach to improve the security of the cipher. *Lecture Notes in Computer Science*, 3017/2004, 2004.

[6] L. Qiao and K. Nahrstedt. A new algorithm for MPEG video encryption. In *CISST*, 1997.

[7] L. Qiao and K. Nahrstedt. Comparison of MPEG encryption algorithms. In *Int. J. on Computer & Graphics*, 1998.

[8] A. Tosun and W. Feng. Lightweight security mechanisms for wireless video transmission. In *Int. Conf. on Information Technology: Coding and Computing*, 2001.

[9] D. S. Turaga, A. A. E. Al, C. Venkatramani, and O. Verscheure. Adaptive streaming over enterprise networks. In *Int. Conf. on Multimedia and Expo*. IEEE, July 2005.

[10] C. Venkatramani, P. Westerink, O. Verscheure, and P. Frossard. Securing media for adaptive streaming. *ACM Multimedia*, Nov. 2003.

[11] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin. A format compliant configurable encryption framework for access control of video. *IEEE Transactions on Circuits & Systems for Video Technology*, 2002.

[12] M. Wu, S. Ma, and W. Shu. Scheduled video delivery - a scalable on-demand video delivery scheme. *IEEE Transactions on Multimedia*, 8(1), Feb. 2006.

[13] XviD.org. ISO MPEG-4 compliant video codec. publicly available at xvid.org.

[14] W. Zeng and S. Lei. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia*, 5(1), Mar. 2003.

[15] W. Zeng, X. Zhuang, and J. Lan. Network friendly media security : rationals, solutions, and open issues. In *Int. Conf. on Image Proc.* IEEE, Oct. 2004.