Efficient and Fully Scalable Encryption for MPEG-4 FGS*

Chun Yuan[1]*, Bin B. Zhu*[2]*, Yidong Wang*[3]*, Shipeng Li*[2]*, Yuzhuo Zhong*[1]
{binzhu,spli}@microsoft.com

[1]Dept. of Computer Science, Tsinghua Univ., Beijing, 100084, China
[2]Microsoft Research Asia, 3F Sigma Center, Haidian District, Beijing, 100080, China
[3]Dept. of Comp. Sci. & Tech., Beijing Univ., Beijing, 100871, China

***Abstract**: The newly adopted MPEG-4 Fine Granularity Scalability (FGS) video coding standard offers full scalability to enable easy and flexible adaptation to changing constraints and different requirements. Encryption of an FGS stream should preserve the full scalability. In this paper, we propose a novel and low complexity scheme to encrypt MPEG-4 FGS streams which enables full FGS functionalities. The encrypted FGS stream can be processed by middle stages directly on the ciphertext without decryption. In addition, the proposed scheme has no degradation on either FGS compression efficiency or error resilient performance, and allows random access. Experimental results as well as a preliminary security analysis of the proposed scheme are also included in this paper.*

Keywords: Video encryption, multimedia security, MPEG-4, FGS, scalable, error resilience, DRM.

1. Introduction

Scalable video coding has gained a wide acceptance due to its flexibility and easy adaptation to a wide range of application requirements and environments. MPEG-4 has recently adopted a scalable video coding scheme called Fine Granularity Scalability (FGS) as a standard [1]. Two profiles have been adopted. One is called Advanced Simple Profile (ASP) which provides a subset of non-scalable video coding tools to achieve high coding efficiency for the base layer. In many applications, the base layer is encoded at very low bit rate. The other is called the FGS profile which is used to obtain the enhancement layer to achieve optimized video quality at a wide bitrate range with the same stream. They will be referred as MPEG-4 FGS in this paper.

Encryption of video data for digital rights management (DRM) has been actively studied and developed in the past decade. There already exist several commercial DRM products in the market now. A typical one is the Microsoft's DRM product for the Windows Media. With appearing of the scalable video coding, it is naturally to require video encryption for this new video format. In addition to the challenges we saw for the encryption design for non-scalable video streams, there are a few more that are specific to FGS. MPEG-4 FGS offers full scalability, and compression is executed only once for a video sequence. When it is delivered to an end user, the stream can be processed by many middle stages to maximize the received quality with resources available. A typical operation by a middle stage is bitrate reduction by simply discarding less important video data. An encryption for FGS should allow rate shaping operations directly on the encrypted video without decryption/re-encryption to reduce the processing load for middle stages, and more importantly, to increase the system security since these middle stages do not need to share any secrets. Encryption system should also preserve the fine granularity in FGS so the remaining bitstream after truncation is still a rate-distortion (RD) optimized video stream. For security reasons, encryption normally works on a large chunk of data. A single wrong bit in a ciphertext renders the whole decrypted plaintext useless. This error propagation feature in encryption may have negative impacts, if not designed properly, to FGS error resilience to transmission errors and packet losses, and to what operations a middle stage can perform on the encrypted FGS stream.

Many algorithms have been proposed for non-scalable video encryption, but we can only mention a few here due to the paper size limitation. Shi et al. proposed to pseudo-randomly change the sign bits of all DCT coefficients [2] or the sign bits of differential values of both DC coefficients and motion vectors [3]. Qiao et al. [4] did a nice description and comparison of some other algorithms. Schemes for scalable multimedia formats have also been reported recently. Wee et al. [5] proposed to encrypt MPEG-4 FGS video data in both base and enhancement layers except header data. Hints for RD-optimal cutoff points are inserted into the unencrypted header for a middle stage to perform RD-optimal bitrate reduction without decryption. Encryption is applied to each packet, which means that the packet size has to be known at encryption, and cannot change after encryption is done. This is undesirable for an encrypted stream to transmit over both wireless networks and internet since wireless transmission requires small packets while internet transmission requires large packets. It also degrades the original FGS fine granularity to packet size. In addition, if any bit in a packet is wrong in transmission or storage, the whole packet will be useless. Grosbois et al. [6] proposed to pseudo-randomly flip sign bits of wavelet coefficients in high frequency subbands for image encryption of JPEG 2000. A different seed is used for each code-block in generating the pseudo-random sequence. These seeds have to be inserted into the compressed stream to send to an end user which lowers the compression efficiency.

The major contributions of this paper are: 1) we report that encryption of the base layer alone is not enough for MPEG-4 FGS due to information leakage; 2) propose an efficient scheme to encrypt base and enhancement layers differently to preserve the original FGS fine granularity and error resilience. In addition, our encryption algorithm is based on the fast algorithm proposed by Jakubowski et al. in which a hash value of the data to be encrypted is used as part of the key to encrypt the data itself. This means that a global key can be repeatedly used to encrypt data chucks in an FGS stream without sacrifice of security. Our scheme does not add bits to the original FGS stream (if the negligible number of bits added to FGST is excluded, see details in Sect. 3.2.2).

2. Background of MPEG-4 FGS

The basic idea of MPEG-4 FGS is to code a video sequence into a base layer and an enhancement layer. The base

*This work was done when Chun Yuan and Yidong Wang were interns at Microsoft Research Asia.

layer uses non-scalable coding to reach the lower bound of the bit-rate range, usually at very low bitrate. The residue of each frame is encoded in the enhancement layer in a scalable manner: the DCT coefficients of a frame's residue are compressed bit-plane wise from the most significant bit to the least significant bit.

It is also possible to use bit-plane coding for the entire DCT coefficients in the temporal enhancement frame, which is called the FGS temporal scalability (FGST). Each FGST VOP can be coded using forward or bi-directional prediction from the base layer. Resynchronization markers can be inserted into FGS stream. The bit-plane separator **fgs_bp_start_code** in the enhancement layer can also be used as the resynchronization marker for error resilience purpose. The base layer and the enhancement layer are unequally protected. The base layer is normally well protected against bit errors or packet losses in real applications. FGS provides very fine granularity scalability to allow near RD-optimal truncations.

3. Efficient and Fully Scalable Encryption

3.1. Base Layer Encryption Is Not Enough

From the structure of MPEG-4 FGS, it is intuitive to apply a non-scalable video encryption algorithm to encrypt only the FGS base layer in FGS encryption. Since the enhancement layer uses the VOPs from the base layer as references, protection of the base layer seems to be enough for the protection of the whole FGS stream. If we look at each individual VOP, such an intuitive thinking is valid. Without knowing the decryption key, a reconstructed frame from both encrypted base layer and unencrypted enhancement layer does not contain any useful information. If we look at a video sequence that consists of VOPs, the picture is very different.

We have done the following experiment to exam if base layer encryption for FGS is enough: we first compress a QCIF video into base layer at low bitrate (around 50 kbps) and enhancement layer with only FGS VOPs at bitrate around 1.0 Mb/s. We then set all the pixel values from the base layer to 0 (or other fixed values), and play the reconstructed FGS video. Based on the test on several typical video clips, we have the following interesting observations: if we exam each individual reconstructed frame, it is very random, and we could not extract any useful information. If we play the reconstructed video, the outline of a moving object large enough is readily visible. More importantly, such a moving object and its action are observed semantic-correctly by human observers. In many applications, such an information leaking in a video encryption system is not acceptable. This phenomenon can be explained by the strong correlation among neighboring frames in video. When the reference frames in the base layer are strongly correlated, a series of enhancement layer frames reveal quite some information. We conclude this subsection with the conclusion that, in contrary to intuitive believing, encryption of the base layer alone is not enough in general for MPEG-4 FGS encryption.

3.2. The Proposed Scheme

The proposed scheme encrypts both the base layer and the enhancement layer of an MPEG-4 FGS stream with different methods. The base layer can be either selectively or fully encrypted. Our full encryption encrypts only video data with VOP headers unencrypted. It is different from the naive encryption mentioned in [4]. The enhancement layer is always selectively encrypted by XORing sign bits of DCT coefficients in the enhancement layer with a pseudo-random sequence. Different frames will use different random sequences to avoid repeatedly applying the same random sequence to different frames, which is the biggest vulnerability of security for a stream cipher. For FGST, the motion vectors are also encrypted in the similar manner as the DCT coefficients.

3.2.1. Encryption of Base Layer

The cipher used for the base layer encryption is the Chain and Sum (C&S) encryption proposed by Jakubowski et al. [7] with minor modifications. In C&S encryption data to be encrypted is first divided into blocks, and then two linear functions are applied to calculate a reversible pre-MAC (Message Authentication Code). Last part of the data is replaced with the encrypted pre-MAC, called MAC. The pre-MAC value is used together with the encryption key to feed into a stream cipher to encrypt the rest data. Division into blocks may result in a partial block at the end. In this case, we use complete blocks to calculate pre-MAC, and apply the stream cipher to the partial block at the end. The number of bits in the partial block is also fed as part of the key into the stream cipher. In this way, the ciphertext has exactly the same size as the plaintext. Since the MAC is reversible, the encryption process can be reversed to get the original plaintext if no bit error occurs. A key feature of C&S encryption is that the stream cipher key depends on a "hash" value of the data to be encrypted. Thus a small difference in the plaintext results in a very different ciphertext, even when the same global encryption key is used. The encryption is very fast [7]. In our implementation, RC4 [8] is used as the stream cipher, and the pre-MAC is encrypted by RC5 [8], even though it is known that one in every 256 RC4 key is "weak" [9].

The base layer can be encrypted either selectively or fully. In the selective mode, the DC values of known number of bits (i.e. **intra_dc_coefficient** and **dct_dc_differential**), sign bits of DCT coefficients, and sign bits of motion vectors **horizontal_mv_data** and **vertical_mv_data,** as well as the motion vector residues **horizontal_mv_residual** and **vertical_mv_residual** are extracted to form a vector to be encrypted with the C&S encryption for each frame. The ciphertext is then put back into the original fields to replace the original data. We note that the number of bits for each encrypted field is either known or can be derived from other unencrypted field. In the full encryption mode, the entropy-coded video data in the base layer except VOP headers is encrypted with the C&S encryption for each frame. For the unlikely case that a ciphertext emulates the start header of the next frame, a header is inserted to the unencrypted VOP headers to indicate the start of the next frame.

We note here that the base layer encryption in either mode does not affect the entropy coding of the video data. Since C&S encryption preserves data size, the proposed encryption scheme has no negative impact to FGS compression efficiency. It is worth noting that the selective encryption scheme is fully MPEG-4 FGS compatible so a standard FGS player can play a selectively encrypted video (although almost useless). This may be advantageous over the full encryption mode that the encrypted video or input of a wrong decryption key may crash a standard FGS player.

It is clear that the proposed scheme has to wait until all the encrypted data has been collected for a frame to perform decryption. This requires buffering the whole base layer frame. It has also to remember the position and number of bits for each encrypted field or to parse the data of a base layer frame twice (one for decryption, and the other for decompression). This may not be acceptable for some applications. If that is the case, we can use a stream cipher such as RC4 to replace the C&S encryption for either of the modes. A different random number is generated for each frame. The number is inserted into the frame header of the base layer. It combines with the encryption key as input to the stream cipher to generate different random bits for each frame to XOR with the data to be encrypted for the frame. For normal length of video, a 32 bit random number should be enough for each frame, and thus the overhead to the compression efficiency is 32 bits per base layer frame. It guarantees non-repeating of random sequences to different frames in a video sequence (unless there is a collision of the random number whose probability is very small if a good random number generator is used). Since RC4 has enormous possible states (around 2^{1700}) [8], it can ensure the security of the encrypted content. The inserted random number is used as the pre-MAC value in the C&S encryption for the enhancement layer encryption described next.

3.2.2. Encryption of Enhancement Layer

The encryption of the enhancement layer is much simpler than the base layer encryption. The pre-MAC value from the base layer an enhancement layer VOP is based on is used together with the encryption key as input to RC4 to generate a random sequence to XOR the sign bits of the DCT coefficients of the VOP in the enhancement layer. Some other items such as a fixed string “Enhancement layer” should also be used as part of the key to RC4 to ensure that the random sequences generated by RC4 for both base layer frame and the enhancement layer frame are different.

The random bits generated by the RC4 are organized into a binary matrix of the same size as the video frame size. The sign bit of a DCT coefficient in the enhancement layer, should it appear in the enhancement layer, would XOR the random bit at the same position of the random binary matrix. In this way, if there is any packet loss, we can easily align the received packets with the correct random bits.

For FGST VOPs, motion vectors are also encrypted. The bits of motion vectors to be encrypted are the same as we described for base layer selective encryption. These bits are encrypted by RC4 in the same way as the DCT sign bits. Several FGST VOPs may use the same base layer frame as the reference. The above described method will generate the same random bits for different FGST VOPs, which is very undesirable for security. To avoid this problem, a frame count is inserted into the FGST VOP header. The count is input to RC4 so different random bits are generated for the FGST VOPs. The frame count can be reused for different group of FGST VOPs since the other part of the RC4 key, i.e., the pre-MAC from the base layer it uses as reference, should be different. Two to three bits should be enough for the frame count in most applications. That means the overhead is about 2 to 3 bits per frame for FGST enhancement layer. This inserted frame count is better than the time stamp in a frame of enhancement layer since some typical operations by a middle stage may change the time stamp. In addition, if some of FGST VOPs are also dropped, the decryption program may not be able to generate the right random bits for the remaining FGST VOPs.

It is easy to see that the proposed enhancement layer encryption preserves the error resilience capability of the original enhancement layer. It enables all the operations on the enhancement layer provided by the FGS compression. Since the base layer is normally protected against packet losses and bit errors in most applications, and all rate shaping or other operations are done on the enhancement layer, we conclude that full FGS functionalities are preserved in our encrypted stream.

3.3. Security of Proposed Scheme

In evaluating the security of a video encryption, two different aspects need to be considered. One is the visual effect, the other is the system security. Visual effect means how much useful information a user can perceive with encrypted video. System security is the same as the traditional encryption system. It indicates how secure a system is under passive and active attacks. Unlike text encryption, a video encryption may contain strong correlation in the plaintext video that an attacker can exploit. The judgment on a successful attack is also different. In text encryption, part of text has to be completely recovered to claim a successful attack. For video encryption, an approximate recovery of the original video may render the system failure, due to irrelevancy exists in most video clips. Different applications may have different criteria for security failure.

As we will see in the next section, the selective encryption mode still leaves some visible structures in the encrypted video. It is also easy to tell the difference between static parts and moving parts. The full encryption model, on the contrary, removes any structures in the original video. The encrypted video is random and useless.

As for the system security, both modes are robust to known-plaintext attack, thanks to the content-dependent input to RC4. The robustness to other attacks is quite different for the selective and full encryption modes. For the full encryption modes, the security depends on the underlying cipher used for encryption. It should be reasonably secure with the C&S encryption used. The selective encryption, on the other hand, may be much less secure, due to limited number of states.

As Tang [10] showed, encryption of DC coefficients alone leaves edges of an encrypted frame still comprehensible. Encryption of sign bits for AC coefficients can only use non-zero AC coefficients. We have done an experiment to count non-zero AC coefficients of I-blocks in the base layer for QCIF videos. When AC coefficient prediction is turned on, Miss America has 1 non-zero AC coefficient per 8 by 8 block on the average at 30kbps for the base layer, and 4.3 non-zero AC coefficients for the Coast Guard at 100 kbps. Tests on other video clips resulted in similar numbers. This means that a brute force attack on the sign bit encryption of AC coefficients requires 2 tries for Miss America and around 16 tries for Coast Guard on average for an 8 by 8 block. It does not take too long to crack the sign bit encryption for AC coefficients.

Since the number of motion vectors for a frame is limited, encryption on the sign bits of motion vectors is not very secure under brute force attack. The encryption on motion vector residues does not help much for security since the number of bits used for motion vector residue is usually very small (under 2 bits), and a wrong motion vector residue does not cause much perceptual damage.

We conclude that the selective encryption, although offers full compatibility with MPEG-4 FGS, does not offer high security. Since we focus on entertainment applications in which video normally has low value. As long as the video encryption system costs an attacker more than buying the video, it should be enough. The selective encryption mode is still a viable solution for many applications.

4. Experimental Results

The proposed scheme has been implemented in pure C++ codes. The C&S encryption was implemented on the field $Z(2^{31}-1)$ which has the security of 2^{62} [7]. All tests were done on a PIII 667 MHz Dell PC with 512MB memory. The encryption/decryption speed (including RC4 and RC5 operations) for our implemented C&S encryption is about 90 Mb/s, which is much slower than the results reported in [7]. The big difference in speed might be due to lack of optimization, fine tune, and assembly codes in our implementation.

Table 1: Base layer bitrates and encryption speeds for the selective and full encryption modes. (Base means base layer only, and all means base layer plus enhancement layer).

	Base layer Bitrate (Kb/s)	Selective (Mb/s)		Full (Mb/s)	
		Base	All	Base	All
Akyio	7.65	25	7905	55	17391
Carphone	24.2	29	3122	82	8828
Coastguard	27.2	31	2978	86	8261
Foreman	32.2	30	2400	90	7200
MissAm	8.62	31	9257	55	16423
Salesman	10.4	29	7201	59	14650

Table 1 lists the base layer bitrates and the encryption speeds for both selective and full encryption modes. The base layer frame rate was one third of the original rate, and the bitrates for the enhancement layer were all at 2.5Mb/s. The number of bits in the base layer processed by the selective encryption mode ranged from 12.67 to 15.72% of the total bits in the base layer. As we can see from Table 1, the selective mode is always slower than the full mode, due to the fact that extraction of encryption bits and putting them back after encryption took more time than direct encryption by C&S encryption. With a fast cipher, a full encryption may be faster than selective encryption. A full encryption is also more secure in general.

Visual effects for both modes are shown in Figure 1 for Akyio and Foreman. As we can see, some of the outline of the speaker for the selective mode is partial visible, while the full encryption renders the resulting frame completely random.

5. Conclusion

We have proposed an efficient encryption scheme for MPEG-4 FGS encryption which enables full functionalities and features of FGS formats. Encryption is executed once and middle stages can process the encrypted FGS stream directly without decryption. The proposed scheme incurs no degradation on either FGS compression efficiency or its error resilience performance.

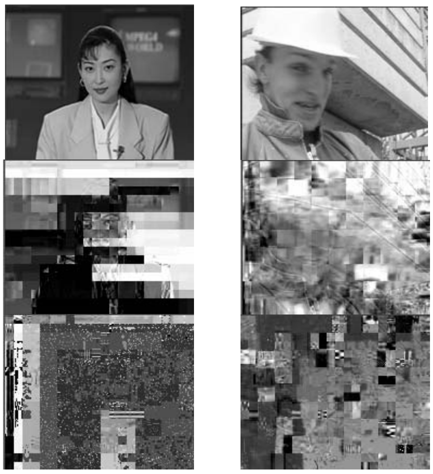

Figure 1. Encryption visual effect. Top: the original. Middle: selective encryption. Bottom: full encryption.

Acknowledgement

The authors would thank our colleagues Feng Wu, Guobin Shen, Xiaoyan Sun, and Jizheng Xu for valuable comments, suggestions, and fruitful discussions.

References

[1] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 11, No. 3, March, 2001, pp. 301--317.

[2] C. Shi and B. Bhargava, “A Fast MPEG Video Encryption Algorithm,” *Proc. of ACM Multimedia’98, 1998*, pp. 81--88.

[3] C. Shi and B. Bhargava, “An Efficient MPEG Video Encryption Algorithm,” *IEEE Proc. 17th Symp. Reliable Distributed Systems, 1998*, pp. 381--386.

[4] L. Qiao and K. Nahrstedt, “Comparison of MPEG Encryption Algorithms,” *Int. Journal on Computers & Graphics, Special Issue: "Data Security in Image Communication and Network"*, Permagon Publisher, Vol. 22, No. 3, 1998.

[5] S. J. Wee and J. G. Apostolopoulos, “Secure Scalable Streaming Enabling Transcoding Without Decryption,” *IEEE Int. Conf. Image Processing*, October 2001.

[6] R. Grosbois, P. Gerbelot, and T. Ebrahimi, “Authentication and access control in the JPEG 2000 compressed domain,” *Proc. of SPIE 46th Annual Meeting, Applications of Digital Image Processing XXIV*, San Diego, 2001.

[7] M. H. Jakubowski and R. Venkatesan, “The Chain & Sum Primitive and Its Applications to MACs and Stream Ciphers,” *EUROCRYPT’98*, pp. 281--293, 1998.

[8] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd ed., John Wiley & Sons, Inc. 1996.

[9] A. Roos, *A Class of Weak Keys in the RC4 Stream Cipher*, attachment to e-mail to cipherpunks@toad.com, September 22, 1995.

[10] L. Tang, “Methods for Encrypting and Decrypting MPEG Video Data Efficiently,” *Proc. ACM Multimedia’96*, 1996, pp. 219--230.