Layered Access Control for MPEG-4 FGS Video[1]

Chun Yuan[1], Bin B. Zhu[2], Ming Su[3], Xiaoming Wang[3], Shipeng Li[2], Yuzhuo Zhong[1]

[1]Dept. of Computer Science, Tsinghua Univ., Beijing, 100084, China
[2]Microsoft Research Asia, Beijing, 100080, China
[3]School of Mathematics, Nankai Univ., Tianjin, China

ABSTRACT

MPEG-4 has recently adopted the Fine Granularity Scalability (FGS) video coding technology which enables easy and flexible adaptation to bandwidth fluctuations and device capabilities. Encryption for FGS should preserve such adaptation capabilities and allow intermediate stages in the delivery to process the media on the ciphertext directly. In this paper, we propose a novel scalable access control scheme with this property for the MPEG-4 FGS format. It offers free browsing of the low-quality base layer video but controls the access to the enhancement layer at different service levels based on either PSNR or bitrates. Both types of service levels are supported simultaneously without jeopardizing each other's security. The scheme is fast and degrades neither compression efficiency nor error resilience of the MPEG-4 FGS. The approach is also applicable to other scalable multimedia.

1. INTRODUCTION

Scalable video coding has gained increasing 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 the Fine Granularity Scalability (FGS) [1] as a standard. In MPGE-4 FGS, a video stream is divided into two layers: a base layer and an enhancement layer. The base layer encodes a video sequence at a very low bitrate in a non-scalable way. The residue between the original video and the reconstructed base layer is then is encoded as the enhancement layer in a scalable manner: DCT coefficients of a frame's residue are compressed bit-plane-wise from the most significant bit to the least significant bit. A video sequence is compressed by MPEG-4 FGS only once. During the process that an FGS media is delivered to an end user, the media can be processed by many intermediate stages to maximize the received quality with available resources. A typical operation by an intermediate stage is rate shaping or transcoding such as bitrate reduction by discarding less important data in the enhancement layer to adapt to narrowed network bandwidths.

Encryption of video data for digital rights management (DRM) has been actively studied and developed in the past decade. A challenge to the DRM design for the MPEG-4 FGS is to preserve the scalable feature after encryption so intermediate stages can process directly on the encrypted video without decryption. This property is desirable in many applications. Such a DRM system would dramatically leverage the system security since no secret needs to be shared with any intermediate processing stages. The processing load on an intermediate stage is also reduced since no decryption and re-encryption cycle is needed in processing a video stream. In addition, scalable video coding makes scalable DRM possible, which non-scalable formats cannot offer. In scalable DRM, the *same* bitstream offers access to different service levels, and the video data at lower service levels can be reused by a higher service level. To support access to a video sequence at different quality levels in non-scalable coding, the same video would have to be compressed into separate files of different qualities, with each file encrypted with a different encryption key. Video data at different levels cannot be reused in this case. When designing a scalable DRM, it is a great challenge to support service levels based on both PSNR and bitrates simultaneously in the same bitstream without jeopardizing each other's security. It is also a challenge to design an encryption scheme with no or minimum degradation to the original system's compression efficiency and resilience to transmission bit errors and packet losses.

While there are many proposed encryption algorithms for non-scalable multimedia formats [2][3], some of which such as the scrambling algorithms proposed in [3] are also applicable to scalable formats, we have seen only a couple of recently reported schemes that were specifically designed for scalable multimedia. Wee et al. [4] proposed a secure scalable streaming (SSS) scheme for scalable coding that enables transcoding without decryption. For MPEG-4 FGS, SSS encrypts video data except headers in both base and enhancement layers. Hints for RD-optimal cutoff points are inserted into the unencrypted header for an intermediate stage to perform RD-optimal bitrate reduction. Encryption granularity depends on how a video stream is packetized in transmission. More precisely, encryption is applied to each transmission packet. This means that SSS encryption has to know the size of transmission packets. Any change to the size of transmission packets will require a cycle of decryption and re-encryption. It is thus similar to the encryption applied at transmission. Grosbois et al. [5] proposed an encryption algorithm for scalable image compression JPEG 2000. In the scheme, sign bits of wavelet coefficients in high frequency subbands are pseudo-randomly flipped. 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. We have recently proposed an encryption scheme for MPEG-4 FGS [6] which encrypts the base layer and also the sign bits of DCT coefficients in the enhancement layer to enable full scalability for the encrypted video.

In this paper, we present a novel scalable access control scheme for the MPEG-4 FGS which encrypts the FGS enhancement video data (headers are not encrypted) at different access levels called *service levels*. Service levels can be divided

[1] The work was done when Chun Yuan, Ming Su, and Xiaoming Wang were interns at Microsoft Research Asia.
Contact info: binzhu@microsoft.com.

according to the PSNR (called *PSNR service levels*) or bitrates (called *bitrate service levels*). PSNR service levels are a natural choice if we separate service levels according to different perceptual qualities, although it is well known that PSNR is not a good measure of perceptual quality. If video is streamed over a network, it is naturally to separate service levels by bitrates. A single type of service levels does not work well for both scenarios since each frame may have variable number of coded bit-planes or bits. The goal of this project is to design a scalable DRM system that allows free views of low-quality video sequences but protects better quality content at different quality levels, with higher level reusing the data of lower levels. Moreover, both PSNR service levels and bitrate service levels have to be supported in the same encrypted video to address a large variety of applications. Our proposed scheme meets all these requirements. Moreover, it has a very low complexity, and causes no degradation to either compression efficiency or resilience to transmission bit errors and packet losses. The same approach can be easily applied or extended to other scalable multimedia formats.

2. THE PROPOSED SCHEME

In MPEG-4 FGS, video data is grouped into video packets, which are separated by the resynchronization marker. The bit-plane start code, *fgs_bp_start_code*, also serves as a resynchronization marker for error resilience purpose [1]. For our purpose in this paper, both the resynchronization marker and the bit-plane start code will be referred to as *vp_marker*, and the data separated by a *vp_marker* will be called a *video packet*. Video packets are aligned with macroblocks. The size of a video packet may vary greatly because the most significant bit-plane may have much less bits than others, and also because a bit-plane may be grouped into video packets such that the last video packet of that bit-plane is much smaller than others. The latter case can be avoided by adjusting video packet sizes of that bit-plane, and we can assume it will not occur in our applications. Note that in the MPEG-4 FGS, video packets are determined at the time of compression. Although resynchronization markers can be moved around, removed or inserted after compression in the MPEG-4 FGS, we assume that video packets don't change after compression in our applications.

For multimedia transmission, we will make the following assumption in our design:

- A transport packet should contain complete video packets so a whole video packet is either received or completely lost when a packet loss occurs.
- The header of a transport packet should contain the information of the indexes of the video packets it contains. When a video packet is received, it can be decompressed to right positions no matter previous video packets are lost or not during transmission.

These two assumptions are valid for most multimedia networks in real applications. We note that the current macroblock address, *fgs_macroblock_number*, is inserted with the resynchronization marker so a received video packet can be decoded correctly no matter previous video packets are received or not.

Ideally we would like the bitrate control as fine as one byte. This is not necessary since each transmission packet contains complete video packets, as we have assumed above. By adjusting the number of transmitted video packets for each frame, the bitrate can be controlled within the deviation of one video packet per second, which is very smaller as compared to the normal video transmission bandwidth. Therefore each video packet can be treated as an encryption cell which is encrypted independently. If the number of bits in the most significant bit-plane is too smaller, it will be combined with the next video packet to form a larger encryption cell. A custom marker *merged_vp* is inserted into the enhancement layer header of the frame to indicate such a situation (see below). In our experiments, only the most significant bit-plane may need to merge into the next video-packet, so one bit per frame for *merged_vp* will be adequate.

A PSNR service level is a group of adjacent bit planes of enhancement layer data, and a bitrate service level is a group of adjacent video packets. A content owner can specify where to separate a PSNR or bitrate service level according to the characteristics of the video and business needs, which is beyond the scope of this paper. Suppose we divide the total bit planes into T adjacent groups to form T PSNR service levels, and divide a frame's enhancement video packets into M adjacent groups to form M bitrate service levels, we have then divided the FGS enhancement layer into $T \times M$ different segments. If the separation point for a bitrate service level coincides with that of a PSNR service level, the corresponding segment is considered as empty with length of 0. Each segment is assigned an independently and randomly generated key denoted as $Key(t, m)$, where $t = 1, \cdots, T$, and $m = 1, \cdots, M$.

Each encryption cell in a segment is encrypted independently with the same segment key. The cipher used to encrypt each encryption cell is based the C&S encryption proposed by Jakubowski and Venkatesan [7] where RC4 [8] is used as the stream cipher and RC5 [8] replaces the original DES to encrypt the pre-MAC. The key idea behind the C&S encryption scheme is to partition the data to be encrypted into blocks and then apply two linear functions to these blocks to obtain a reversible pre-MAC (Message Authentication Code) which replaces some data blocks. The resulting pre-MAC value is combined with the encryption key to feed into RC4 to encrypt the transformed data other than the pre-MAC as well as the trailing partial block, if it exists, while the pre-MAC itself is encrypted by RC5 to form the MAC. Since the MAC is reversible, the encryption process can be reversed to get the original plaintext if no bit error occurs. Due to the unique feature that part of the key to the stream cipher is the "hash" value of the data to be encrypted, a small change in the data to be encrypted will result in very different ciphertext even though the same global encryption key may be repeatedly used. Another advantage of the C&S encryption is that it does not increase the size of the data to be encrypted. In our implementation, the C&S encryption is implemented on the field of $Z(2^{31}-1)$ which has the security of 2^{62} for each encryption cell [7]. For details of the C&S encryption, interested readers are referred to [7].

Our system focuses on the superdistribution model where the content and the license are delivered separately, although the scheme can be equally applicable to other distribution models. When a user buys a certain service level, all the keys for that and lower levels of the same service type will be included in the license delivered to the user. For example, if the service level a user buys is the PSNR service level $t = 2$, the license will contains all the keys $Key(t, m)$, where $t <= 2$, and $m = 1, \cdots, M$. It is similar for the case of bitrate service levels. In this way, a user with access to a certain type of service level can only access that service level and lower service levels of the same type. Access to higher service levels of the same type or service levels of different type is not granted. Note that only a subset of segment keys are likely used for encryption in most applications since there exists correlation between PSNR service levels and bitrate service levels. For example, a low PSNR service level is likely to share data with a low bitrate service level. This means that some of the total $T \times M$ segments are likely to be empty (i.e., of length 0). Since we use the superdistribution model, the unused keys will have no impact to the content encryption. It affects only the size of a license. If the size of a license is a concern for an application, we can exclude unused keys from the license. A scheme that requires only a single key in a license to send to a buyer is to be reported elsewhere [9].

No marker is needed to separate PSNR service levels since the separation points for all frames are the same, and MPEG-4 FGS bit-plane start code can be used for such a purpose. Separation points for bitrate service levels vary from one frame to another, but no marker is actually inserted into the bitstream to separate each bitrate service level for a frame. Instead a custom header is added to the enhancement layer of each frame to indicate how many video packets the enhancement layer has for the frame at the encryption time. It also uses a few bits to register possible minor variation from a general grouping rule to allow fine tune of bitrate service levels for the frame. The previous mentioned marker *merged*_vp is placed into this header, too. For our experiments, 24 bits will be adequate for such a purpose. In this way, our scheme has negligible compression overhead: only 24 bits are added to each compressed frame. Because of the 2nd assumption at the beginning of this section, the bitrate service level separation points can be derived from this 24 bit customized header even under the circumstance that some video packets are lost in transmission, which guarantees that the right key is used to decrypt each received video packet. In applications where the 2nd assumption is invalid, a field has to be inserted into the unencrypted part for each video packet to indicate which bitrate service level it belongs to. Note that in streaming applications, it may be advantageous for a streaming server to derive bitrate service level separation points so it does waste bandwidth in sending extra data that an end user has no permission to access.

There is a slim chance that the ciphertext emulates the video packet separator *vp_marker*. We adopted the following simple technique to address the issue: If a ciphertext emulates *vp_marker*, the marker *vp_marker* at the beginning of the cell is repeated, followed by a fixed number of bits to indicate how many *vp_markers* to skip (or the size of the encryption cell) before appending the ciphertext. The same technique can be used if the hamming distance between *vp_marker* and any vector in the ciphertext is smaller than a threshold. Since the chance for such an event occurs is very tiny, the byte overhead is negligible.

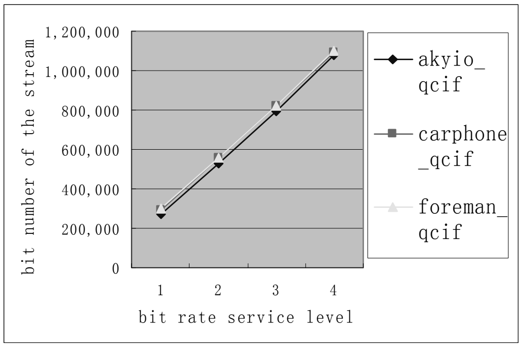


Figure 1. Average number of bits for the four bitrate service levels for *akyio, carphpone,* and *foreman*.

3. PERFORMANCES OF PROPOSED SCHEME

3.1. Error Resilience

First we discuss the performance of the proposed scheme under transmission bit errors and packet losses. Let us begin by looking at its performance when packet loss occurs. With the two assumptions at the beginning of Section 2, any received packet can be fully decrypted no matter previous packets are lost or not, as long as the packet containing the custom header for the frame mentioned previously has received. If the packet containing the custom header has lost, the frame headers are also lost since they are placed together, and thus the whole enhancement layer for the frame has to be discarded, no matter the enhancement layer is encrypted or not. We conclude that the proposed scheme has no degradation under packet losses. In other words, the scheme is robust to packet losses.

In MPEG-4, if bit errors occur in a received bit stream, the video packet that contains bit errors will normally be discarded, if we assume that the Reversible Variable Length Codes (RVLC) are not used. Using RVLC reduces coding efficiency. With our proposed scheme, a bit error in an encryption cell will expand to many other bits inside the cell, but never propagate to other cells, thanks to independent encryption of each cell. Only the corrupted cell will be discarded. Since the encryption cell is the same as the video packet (except possible video packets from the most significant bit-plane), the scheme has no negative impact under bit errors. In conclusion, the proposed scheme is robust to both bit errors and packet losses.

3.2. Processing and Compression Overhead

The C&S encryption is a simple and fast algorithm. We have implemented the algorithm in C++ and tested on a DELL PC with PIII 667 MHZ CPU and 512 MB memory. Its encryption or decryption speed was over 90 Mbits/s. This was much slower than the results reported in [7] due possibly to the fact that we did not use any assembly codes or fine-tune to any specific platforms. The computational overhead for both encryption and decryption is very small as compared with encoding and decoding processes.

In the proposed scheme, encryption is applied after the compression, so it does not affect compression efficiency. The data added to the compressed bitstream is the custom header of

24 bits per frame, plus statistically insignificant overhead for *vp_marker* to be emulated in the ciphertext. For video at normal bit rates, the overhead of the proposed scheme is negligible.

3.3. Other Performance

The proposed scheme enables rate shaping operations such as dynamical reduction to a certain service level of either type on the ciphertext directly, as we discussed in early sections. It also allows random access to or reversal play of video (with a smaller buffer), thanks to the nice property of the C&S encryption algorithm that makes possible independent encryption of a reasonably smaller cell without sacrifice of security.

4. EXPERIMENTAL RESULTS

We have implemented a demo system which integrated the proposed scheme into the MPEG-4 FGS reference codes from MPEG that matches the verification model described in [10]. Quite a few video clips have been tested on the system with four PSNR service levels and four bitrate service levels. Figure 1 shows the average bitrate for each of the four bitrate service levels for the three QCIF video clips *foreman*, *akyio*, and *carphone*. The bitrates are almost along a straight line since we used linear division for the bitrate service levels in our experiments.

The visual effect for the four PSNR service levels for *foreman* is shown in Figure 2, along with the PSNR values for each PSNR service levels. In our experiments we separated the PSNR service levels for each video clip in such a way that each PSNR level showed perceptible improvement over the next lower level.

5. CONCLUSION

We have presented in this paper a novel secure and scalable access control for the MPEG-4 FGS format. The proposed scheme exploits unique features offered by the scalability of the FGS format to deliver a scalable DRM which is much more efficient in storage and transmission. Both PSNR service levels and bitrate service levels are supported simultaneously in the encrypted video. One type of access control does not jeopardize the other type of access control, and a higher level control has full access to the lower levels of the same service type, but not vice versa. The scheme is fast and has negligible byte overhead. Moreover, it is robust to both bit errors and packet losses. The scheme can be applied or extended to other scalable multimedia formats and should be useful in a large variety of applications.

6. ACKNOWLEDGEMENT

The authors would thank our colleagues Feng Wu, Guobin Shen, and Xiaoyan Sun of Microsoft Research Asia for valuable comments, suggestions, and fruitful discussions.

7. 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] 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.

[3] J. Wen, M. Severa, W. Zeng, M. H. Luttrell, and W. Jin, “A Format-compliant Configurable Encryption Framework for Access Control of Video,” *IEEE Trans. Circuits & Systems for Video Technology*, vol. 12, no. 6, 2002, pp. 545 – 557.

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

[5] 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.

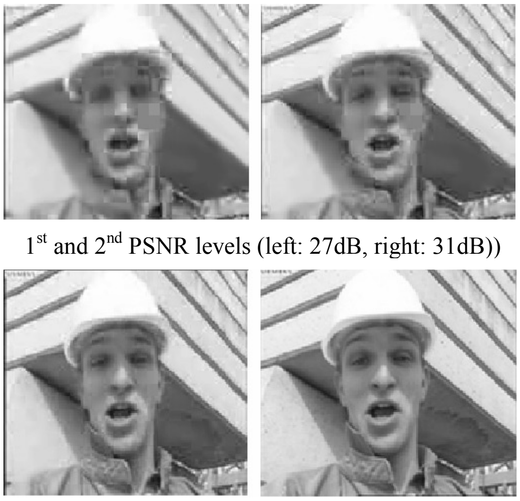
[6] C. Yuan, B. B. Zhu, Y. Wang, S. Li, Y. Zhong, “Efficient and Fully Scalable Encryption for MPEG-4 FGS,” *IEEE Int. Symp. Circuits and Systems*, May, 2003.

[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] B. B. Zhu, C. Yuan, Y. Wang, and S. Li, “Scalable Protection for MPEG-4 Fine Granularity Scalability,” submitted to *IEEE Trans. Multimedia*, Dec. 2002.

[10] ISO/IEC JTC1/SC29/WG11 N3515, July 2000, Beijing.



1st and 2nd PSNR levels (left: 27dB, right: 31dB))

3rd and 4th PSNR levels (left: 36dB, right: 45dB)

Figure 2. Visual effects for the four PSNR service levels for *foreman*.