An Efficient Key Scheme for Layered Access Control of MPEG-4 FGS Video

Bin B. Zhu[1], Min Feng[2], Shipeng Li[1]
[1]*Microsoft Research Asia, Beijing, 100080, P. R. China*
Dept. of Mathematics, Beijing Univ., Beijing, 100871, P. R. China
binzhu@microsoft.com, fengmin@cipher.math.pku.edu.cn, spli@microsoft.com

Abstract

The recently proposed Scalable Multi-Layer FGS Encryption (SMLFE) [4][5] encrypts an MPEG-4 FGS stream into multiply PSNR and bitrate quality layers for layered access control. Both layer types are supported simultaneously. A simple key scheme was used in SMLFE. In this paper, we propose a novel key scheme for SMLFE that reduces the number of keys maintained and managed by a license server for each protected MPEG-4 FGS stream to two. The new key scheme needs only one key contained in a license to be sent to a consumer. This scheme is based on a cryptographic secure hash function and the Diffie-Hellman key agreement. It satisfies all the requirements of SMLFE and can be used to replace the original simple key scheme for SMLFE. The secure one-way hash and intractability of the Diffie-Hellman and the related problems of computing discrete logarithm ensure the security of the new key scheme.

1. Introduction

Scalable coding is a coding technology that encodes a multimedia signal in a scalable manner that a single compressed stream can be easily adapted to a wide range of applications. Several scalable coding formats have recently been adopted by standards organizations: The Joint Photographic Experts Group (JPEG) has recently adopted the wavelet-based JPEG 2000 [1] which is a scalable image coding format. The Moving Picture Experts Group (MPEG) has also adopted a fine grain scalable video coding format called Fine Granularity Scalability (FGS) to its MPEG-4 standard [2]. In this scalable video coding format, a video sequence is compressed into a single stream with two layers: a base layer and an enhancement layer. The base layer is a non-scalable coding of a video sequence at the lower bound of a bitrate range. The enhancement layer encodes the difference between the original sequence and the reconstructed sequence from the base layer in a scalable manner to offer fine grain scalability in a large range of bitrates for the sequence. MPEG-4 FGS enables one–compression–to-meet-the-needs-of-all applications, which is very desirable in many multimedia applications. Rate reduction and other rate shaping operations can be performed directly on a compressed stream without decompression.

Scalability offered by scalable formats enables new services that cannot be offered by non-scalable formats. One of such new services is a layered access control with a single scalable stream. A simple layered access control algorithm that has a single encrypted layer plus an unencrypted layer was proposed in [3] for the scalable JPEG 2000 image coding format. To support access control of different scalabilities such as access control on resolutions or on image qualities in this scheme, the same scalable stream has to be encrypted differently into different encrypted streams. A much more sophisticated layered access control scheme called Scalable Multi-Layer FGS Encryption (SMLFE) was recently proposed to support both PSNR and bitrate layers simultaneously for the MPEG-4 FGS format [4][5]. In SMLFE, a single MPEG-4 FGS stream is encrypted into a single encrypted stream with multiple quality layers partitioned according to PSNR values and bitrates. Both types of quality layers are supported simultaneously so a server can choose a desired layer of either type directly from an SMLFE-encrypted stream without decryption. In SMLFE, lower quality layers are accessed and reused by a higher quality layer of the same type, but not vice versa. The protection of the two different layer types is orthogonal, i.e., a right to access a layer of one type does not make the layers of the other type also accessible. To achieve this goal, SMLFE partitions each enhancement frame into T PSNR layers and M bitrate layers independently. An enhancement frame is therefore partitioned into $T \times M$ different segments, with possible existence of empty segments. Each segment resides in one PSNR layer and one bitrate layer simultaneously. Each segment is encrypted with the

This work was done when Min Feng was a visiting student at Microsoft Research Asia.

corresponding segment key. A segment key is reused to encrypt the same indexed segments of different frames.

Multimedia Digital Rights Management (DRM) manages all rights for multimedia from creation to consumption [6][7]. MPEG has been actively developing a DRM framework, the Intellectual Property Management and Protection (IPMP), for the MPEG-4 standard [8][9]. There are also several commercial DRM products available on the market. A typical one is the Windows Media Rights Manager (WMRM) from Microsoft [10]. Copyright law distinguishes between copyright (the right to copy or distribute) and useright (the right to "perform" or to use a copy once obtained). A DRM system such as WMRM separates distribution of the protected content from that of the decryption key, and controls the usage of digital content rather than its distribution. In fact, DRM-protected content is typically distributed in superdistribution, a powerful distribution mechanism that treats ease of replication of digital content as an asset rather than a liability. Superdistribution actively encourages free distribution of digital content via any distribution mechanism imaginable to reach maximum number of potential consumers. A DRM system such as WMRM encrypts and packages digital media into a digital media file to be distributed in superdistribution. The decryption key is uploaded to a license server along with a specification of rights to use the content desired by the content owner. To play protected content, a consumer Alice or the player she uses first acquires a license from the license server which contains the decryption key as well as the rights Alice has obtained for the content. A license is individualized, typically encrypted with a key that binds to the hardware of Alice's player, so the license cannot be illegally shared with others. Detailed description on processing performed by a DRM system such as WMRM can be found in [11].

Unlike other DRM system, an SMLFE-based DRM system offers preview of the protected content by simply leaving the base layer unencrypted. This is a much desired feature in superdistribution of DRM-protected content. The embedded preview version in SMLFE serves as its own advertisement. It enables a potential consumer to preview the content before buying a right to access the content with a better quality. A switch from one quality layer to a higher quality layer or to a layer of different type is very simple in SMLFE. A consumer only needs to acquire another license with a new set of decryption keys from a license server.

SMLFE proposed in [4][5] encrypts each segment with the corresponding segment key. There segment keys are independently generated. For a partition of T PSNR layers and M bitrate layers, $T \times M$ different segment keys are needed to encrypt an MPEG-4 FGS stream, if there is no rekeying in encryption of the stream. A license server has to maintain and manage all these $T \times M$ segment keys for each encrypted stream. When a consumer Alice acquires a right to access a quality layer, all the segment keys for the segments that Alice has a right to access have to be delivered to her in a license. This increases complexity of key management at both the license server and the consumer client. Workload of a license server and the size of a license have also been increased.

In this paper, we shall propose a novel key scheme for SMLFE which is much more efficient than the key scheme proposed in [4][5]. The new key scheme is based on the Diffie-Hellman key agreement protocol [12] and a cryptographic hash function. Two independent keys are needed for each type of quality layers, respectively. The key of a lower quality layer is the hash value of the key of the next higher quality layer of the same type. The logarithm values of all layer keys are packaged into the protected content. When a consumer Alice acquires a right to access a layer, the key of the selected layer is calculated by the license server and sent in a license to Alice. In other words, a license contains a single key, exactly the same as in a license for a non-scalable stream. The DRM module at the client side uses the layer key contained in the license and the logarithm values packaged inside the protected content to calculate the segment keys for the segments that Alice has a right to access. The Diffie-Hellman key agreement is used in calculation of segment keys. The resulting segment keys are then used to decrypt corresponding segments. This new key scheme is secure and meets all the requirements of SMLFE. It can be used to replace the original key scheme used in SMLFE as described in [4][5].

2. Original key scheme in SMLFE

As a preparation to describe the proposed key scheme in this paper, we first describe the original key scheme for SMLFE described in [4][5]. In SMLFE, a PSNR layer is a group of adjacent bit planes in each enhancement frame. A bitrate layer is a group of adjacent video packets. A video packet in MPEG-4 FGS is a block of video data separated by resynchronization markers or the bit-plane start code. Each layer of either type is aligned with video packets. Suppose that the bit-planes of each enhancement frame are partitioned into T adjacent groups to form T PSNR layers, and that all video packets in an enhancement

frame are partitioned into M adjacent groups to form M bitrate layers, data of an enhancement frame is then partitioned into $T \times M$ different segments $\{S_{t,m}\}$, where $t = 1, \cdots, T$ and $m = 1, \cdots, M$. A large layer number means a higher quality layer. There exists some correlation between PSNR layers and bitrate layers. For example, a low PSNR layer is likely to share data with a low bitrate layer but unlikely to share with a high bitrate layer. This means that some segments out of the total $T \times M$ segments per enhancement frame are likely to be empty (i.e., of length 0). Figure 1 shows an example of segments for an enhancement frame where empty segments are not shown.

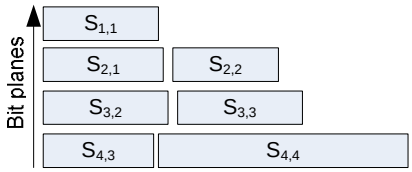

Figure 1: An example of segments for an enhancement frame for $T = M = 4$.

In SMLFE, a set of $T \times M$ different segment keys $\{K_{t,m}\}$ are independently and randomly generated for each video sequence. The key $K_{t,m}$ is used to encrypt the segment $S_{t,m}$ for each enhancement frame, where $t = 1, \cdots, T$ and $m = 1, \cdots, M$. A non-empty segment $S_{t,m}$ is first partitioned into video packets. Video data in each video packet is then encrypted with the segment key $K_{t,m}$ and the encryption algorithm proposed in [13]. Table 1 shows segment keys that match the example given in Figure 1. Note that partition of segments may vary from one enhancement frame to another.

Table 1: Segment keys for $T = M = 4$.

	m=1	m=2	m=3	m=4
t=1	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$
t=2	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$
t=3	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$
t=4	$K_{4,1}$	$K_{4,2}$	$K_{4,3}$	$K_{4,4}$

When a consumer Alice acquires a right to access a certain quality layer, all the keys for that and lower quality layers of the same type are sent in a license to Alice. For example, if the layer Alice has acquired is the PSNR layer $t = 2$, then the $2M$ keys $\{K_{t,m}\}$, where $t \le 2$ and $m = 1, \cdots, M$, are sent in a license to Alice. In this way, a right to access a quality layer can also access lower quality layers of the same type. Higher quality layers of the same type and quality layers of a different type are not accessible.

3. Proposed key scheme

In our new key scheme to be described next, each layer is assigned a key, called a *layer key*. Segment keys used in SMLFE described in Section 2 is generated from layered keys. Layer keys are derived from the layer key of the highest quality of the same type. For simplicity of description, the following notations are described first.

Let H be a cryptographic hash function. $H^n(x)$ denotes the result after the hash function H is applied n times to x, where $H^0(x) := x$. $Z_n := \{0, 1, 2, \cdots, n-1\}$ denotes the integers modulo n where addition, subtraction, and multiplication in Z_n are performed modulo n. $Z_n^* = \{a \in Z_n \mid \gcd(a, n) = 1\}$ denotes the multiplicative group of Z_n. Let p be a prime number and g be a generator of Z_p^*, $2 \le g \le p-2$. Let the t^{th} PSNR layer key be x_t, and m^{th} bitrate layer key be y_m, $t = 1, \cdots, T$ and $m = 1, \cdots, M$. Let $x_T \equiv x$, and $y_M \equiv y$, where x and y are two independent secret random numbers in the range of [1, p-2].

In our new key scheme, layer keys are generated by hashing the layer key of the next higher quality layer of the same type:

$x_t = H^{T-t}(x) \bmod p$, $y_m = H^{M-m}(y) \bmod p$. Eq. (1)

(If any value in Eq. (1), say x_t, is outside [1, p-2], the key x_t used in the calculation of the logarithm value and the segment keys to be described next is actually $x_t + r \bmod p$, where $2 \le r \le p-2$. This is assumed in the following description without being explicitly mentioned.)

For each layer key, x_t or y_m, we calculate its logarithm value $X_t = g^{x_t} \bmod p$ or $Y_m = g^{y_m} \bmod p$. A segment key K_{x_t, y_m} for the segment S_{x_t, y_m} is generated as follows:

$K_{x_t, y_m} = g^{x_t \cdot y_m} = (X_t)^{y_m} = (Y_m)^{x_t} \bmod p$. Eq. (2)

The set of the logarithm values $LVS = \{X_t, Y_m \mid t = 1, \cdots, T; m = 1, \cdots, M\}$ are packaged with the protected content. They will be used in calculating segment keys at a client side.

When a consumer Alice has acquired a right to access some layer, say a PSNR layer b without loss of generality, the layer key x_b for the PSNR layer b is calculated from x with Eq. (1) and is placed in the license to be sent to Alice. At the client side, the layer key x_b contained in the received license is used to calculate all the PSNR layer keys below b with Eq. (1): $x_t = H^{b-t}(x_b), 1 \le t < b$. The segment keys that PSNR layer b can access, $K_{t,m}, 1 \le t \le b, \forall m$, can be calculated with Eq. (2) from the just obtained PSNR layer keys and the logarithm values from the set *LVS* that are packaged in the protected content.

Compared with the original key scheme described in [4][5], the number of keys maintained and managed by a license server is reduced from $T \times M$ keys to two keys in the new key scheme. The number of keys contained in a license for a consumer is also reduced to a single key. If we treat the aforementioned logarithm values also as keys ("public keys"), the total number of keys needed at the client side, either packaged in the protected content or contained in a license, is $M + T + 1$, as compared to potentially $T \times M$ in the original key scheme.

4. Security and implementation issues

In the proposed key scheme, the key of a lower layer is generated by hashing the key of the next higher layer of the same type. Two independent keys are used for PSNR and bitrate layer types. A cryptographic hash function is used in this procedure. This guarantees that a higher layer can access all the lower layers of the same type but not vice versa, and access of one layer type does not gain access to the other type, as required by SMLFE. The security of segment keys generated in the proposed key scheme rests on the intractability of the Diffie-Hellman problem and the related problem of computing discrete logarithm [12]. In fact, the segment key calculation of Eq. (2) is the basic version of the Diffie-Hellman key agreement protocol. We conclude that the proposed key scheme is secure.

In actual implementation of the proposed key scheme, Elliptic Curve Diffie-Hellman (ECDH) key agreement can be used rather than the Diffie-Hellman form of Eq. (2). For example, the ECDH and its parameters proposed in [14] by the Secure Shell Working Group for the SSH transport level protocol can be used in the proposed key scheme for SMLFE.

5. Conclusion

In this paper, we have proposed a new key scheme based on a cryptographic hash function and the Diffie-Hellman key agreement. The proposed key scheme reduces the number of keys maintained and managed by a license server from $T \times M$ in the original key scheme to 2. Only one key is needed to be sent to a consumer in a license, as compared to potential $T \times M$ keys in the original key scheme. The proposed key scheme is secure and satisfies all the requirements of SMLFE.

References

[1] "Information Technology – JPEG 2000 Image Coding System," *ISO/IEC International Standard 15444-1*, ITU Recommendation T.800, 2000.

[2] 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.

[3] R. Grosbois, P. Gerbelot, and T. Ebrahimi, "Authentication and Access Control in the JPEG 2000 Compressed Domain". *Proc. SPIE 46th Annual Meeting, Applications of Digital Image Processing XXIV*, San Diego, California, 2001.

[4] C. Yuan, B. B. Zhu, M. Su, X. Wang, S. Li, and Y. Zhong, "Layered Access Control for MPEG-4 FGS Video," *IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sept. 2003, vol. 1, pp. 517 – 520.

[5] B. B. Zhu, C. Yuan, Y. Wang, S. Li, "Scalable Protection for MPEG-4 Fine Granularity Scalability," to be published in *IEEE Trans Multimedia*.

[6] R. Iannella, "Digital Rights Management (DRM) Architectures," *D-Lib Magazine*, vol. 7, no. 6, 2001.

[7] A. M. Eskicioglu, J. Town, and E. J. Delp, "Security of Digital Entertainment Content from Creation to Consumption," *Signal Processing: Image Communication, Special Issue on Image Security*, vol. 18, no. 4, April 2003, pp. 237 – 262.

[8] "*MPEG-4 IPMP Extension Committee Draft*," ISO/IEC 14496-1:2001/AMD3.

[9] "*Study of FPDAM ISO/IEC 14496-1:2001 / AMD3*," ISO/IEC JTC 1/SC 29/WG11 N5068.

[10] *Windows Media 9 Series Digital Rights Management*, http://www.microsoft.com/windows/windowsmedia/drm.aspx.

[11] *Architecture of Windows Media Rights Manager*, http://www.microsoft.com/windows/windowsmedia/wm7/drm/architecture.aspx.

[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[13] M. H. Jakubowski and R. Venkatesan, "The Chain & Sum Primitive and Its Applications to MACs and Stream Ciphers", *EUROCRYPT'98*, pp. 281–293, 1998.

[14] *Elliptic-Curve Diffie-Hellman Key Exchange for the SSH Transport Level Protocol*, http://www.ietf.org/internet-drafts/draft-stebila-secsh-ecdh-00.txt, Nov. 30, 2003.