AGGREGATING SIGNATURES OF MPEG-4 ELEMENTARY STREAMS

Yongdong Wu

Information Security Laboratory
Institute for Infocomm Research (**I^{2}R**)
21, Heng Mui Keng Terrace, Singapore, 119613
wydong@i2r.a-star.edu.sg

ABSTRACT

A complete MPEG-4 stream consists of many elementary streams which may be generated by different authors. In the scenario of this paper, each author signs his own authentic elementary stream independently, and then an untrusted distributor aggregates these signatures into only one. Based on the unique signature, a client is able to verify the received MPEG-4 stream with the certificates of all the authors other than the certificate of the distributor. In addition, each author can not deny what he has signed even if he is willing to admit a signature on another ES. This aggregated signature scheme is efficient in terms of transmission overhead and verification time since only one signature is processed in the client side.

1. INTRODUCTION

MPEG-4 [1] [2] is an excellent multimedia standard for digital television, interactive graphics applications, and interactive multimedia. To safeguard Intellectual Property in the form of MPEG-4, IPMP [3] defines the hooks for the protection methods after MPEG-4 became an International Standard in 1999. To enhance inter-operability, the newer MPEG-4 IPMPX (MPEG-4 part 13) [4] [5] was finalized recently. It not only enables IPMP-compliant devices to render content within an MPEG-4 terminal, but also provides a framework with normative messages to select and configure the most effective and appropriate tools. Following the principle of IPMP, several works (e.g. [6] - [10]) proposed flexible ways to protect MPEG-4 content in a controlled manner. However, IPMPX does not specify any technology for protecting MPEG-4 stream. To fill in this gap, the present scheme describes one authentic scheme for MPEG-4 stream.

According to MPEG-4 specification, an MPEG-4 stream includes many elementary stream (ES) [1] which may come from different authors. A distributor will aggregate the ESs and deliver the complete MPEG-4 stream to the client. To generate an authentic stream, the distributor may aggregate all the ES and sign on the entire stream. However, this straightforward solution may be not viable since the distributor is not trustworthy in some cases. For example, the distributor may be compromised due to viruses. Alternatively, each author signs on his ES for authenticity, and delivers the ES to the client together with its signature. The client is able to verify the authenticity of the individual ES one by one. With this näive solution, the network overhead and computational cost are linear to the number of individual signatures because each signature has to be transmitted and verified separately.

To reduce both network overhead and computational cost, the present scheme enables a client to verify all the ESs generated by the authors with only one digital signature based on an improved aggregated signature scheme. As a result, the computational cost and communication overhead are very small. Since the present scheme produces only one signature for an MPEG-4 stream no matter how many authors are, it is preferable to manage MPEG-4 streams.

The remainder of this paper is organized as follows. Section 2 introduces the basic preliminary. Sections 3 elaborates the aggregated signature scheme. In Section 4, the performance is addressed in terms of security, computational cost and communication overhead. Section 5 summarizes the paper.

2. PRELIMINARY

2.1. One-way hash function

A hash function takes a variable-length input string and converts it to a fixed-length output string, called a hash value. A one-way hash function, denoted as $\mathcal{H}(.)$, works in one direction: it is easy to compute a hash value $\mathcal{H}(m)$ from a pre-image m; however, it is hard to find a pre-image that hashes to a particular hash value. There are many one-way hash functions, such as MD5[11] and SHA[12].

[1] Elementary Stream is conceived as a flow of data that originates from a single source in the sender and terminates in a single sink at the receiver.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2.2. Aggregated Signature

A digital signature algorithm (e.g. RSA [13] and DSA [14]) is a cryptographic tool for generating non-repudiation evidence, ensuring the integrity as well as the origin of the signed message. To authenticate multiple messages efficiently in terms of the communication overhead and computational cost, Boneh *et al.* [15] proposed a cryptographic primitive called aggregated signature (hereafter referred to as the BGLS scheme) which allows aggregation of multiple individual signatures into one *aggregated* signature. Verification of the unified signature is *equivalent* to verifying individual signatures in terms of security. If any of the messages is tampered, the aggregated signature is regarded as invalid.

2.3. MPEG-4 stream

With regard to Fig.1, an MPEG-4 stream includes many elementary stream (ES) which may come from different sources (authors). For Example, in a TV program, video ES may be generated with a video camera, and text ES may be produced from a editor, and advertisement ES may be inserted by a third company. After receiving all the ESs, the broadcast station will combine all the ESs so as to send a complete program.

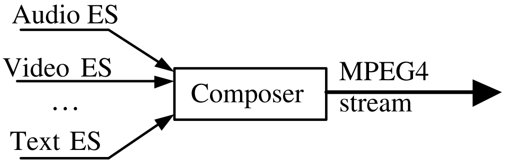

Fig. 1. MPEG-4 stream, where the composer is a module of the distributor.

3. AGGREGATION SCHEME

3.1. System Model

In an authentic MPEG-4 delivery application, there are three kinds of participants: author, client, and distributor. The author is trustworthy and is able to produce his own signature for the specific ES. For instance, the audio source generates the signature for the audio ES. The client receives the MPEG-4 stream, and checks the authenticity of the received stream with the public keys of the authors. The distributor is responsible for combining the authentic ESs and distributing the complete stream. To save network bandwidth and reduce the verification time in the client side, the distributor will also aggregate the authors' signatures into a unique signature. Fig.2 illustrates the production process of an aggregated signature.

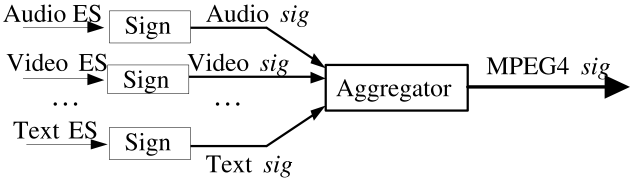

Fig. 2. Aggregating the individual signatures into a unique signature, where the aggregator is a module of the distributo.

3.2. Parameter Configuration

According to the aggregating signature scheme [15], distinct signers generate different signatures on different messages and then all the individual signatures are merged into one short aggregated signature based on elliptic curves [16] and bilinear mappings [17]. A bilinear map is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}$, where:

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are two (multiplicative) cyclic groups of prime order p;
- $\#\mathbb{G}_1 = \#\mathbb{G}_2 = \#\mathbb{G}$, where $\#\mathbb{Z}$ is the cardinality of $\mathbb{Z}$;
- g_1 is a generator of $\mathbb{G}_1$ and g is a generator of $\mathbb{G}_2$.

The bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}$ satisfies the following properties:

1. Bilinearity: for all $\alpha, \beta \in \mathbb{G}_1, \gamma \in \mathbb{G}_2$, $\mathbf{e}(\alpha\beta, \gamma) = \mathbf{e}(\alpha, \gamma)\mathbf{e}(\beta, \gamma)$;
2. Non-degenerate: $\mathbf{e}(g_1, g) \neq 1$

Assume hash function $\mathcal{H}$: $\{0,1\}^* \to \mathbb{G}_1$. To generate a secret key, the key owner picks a random $x \in \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ as his secret key, and computes $v = g^x \in \mathbb{G}_2$ as his public key.Subsequently, the public key will be authenticated by a trusted party.

3.3. Individual signature

After an author P_i produces his own ES m_i, he will sign on the ES to produce an individual signature. To this end, P_i computes the message hash as

$$h_i = \mathcal{H}(m_i \parallel v_i), \tag{1}$$

and the individual signature

$$\sigma_i = h_i^{x_i},$$

where x_i is the secret key of the author P_i. In Eq.(1, the ES m_i is concatenated with the signer's public key. This concatenation results in two advantages: (1) the hash values are different even if two authors produce the same message; (2) the author can not deny what he signed. Afterwards, the original ES m_i and its signature σ_i will be sent to the distributor.

3.4. Aggregating signatures

After each author P_i produces the signature for his ES m_i, and delivers m_i and its signature σ_i to the distributor, the distributor checks the authenticity of each ES with Eq.(**??**) in case of $n = 1$. If any ES is bogus, the distributor rejects the ES. Otherwise, he aggregates all the individual signatures into one signature. Specifically, he computes

$$\sigma = \prod_{i=1}^{n} \sigma_i \tag{2}$$

where σ_i corresponds to the signature on the ES m_i. Because $\sigma \in \mathbb{G}$, the aggregated signature σ is of the same size as an individual signature. The distributor disseminates the complete MPEG-4 stream and the aggregated signature to the clients.

3.5. Verifying signatures

After receiving an MPEG-4 stream and its aggregated signature, the client checks the authenticity of the entire stream before she consumes it. To accomplish this, she sends a request to an authentic database for the public keys or certificates of the authors. Suppose the received ESs are $m'_i, i = 1, 2, \ldots, n$, the client calculates

$$\begin{aligned} h'_i &= \mathcal{H}(m'_i \parallel v_i), \\ \mathbf{e}' &= \prod_{i=1}^{n} \mathbf{e}(h'_i, v_i) \\ &= \prod_{i=1}^{n} \mathbf{e}(h'_i, g^{x_i}) = \prod_{i=1}^{n} \mathbf{e}((h'_i)^{x_i}, g) \\ &= \prod_{i=1}^{n} \mathbf{e}(\sigma'_i, g) = \mathbf{e}(\prod_{i=1}^{n} \sigma'_i, g) \end{aligned} \tag{3}$$

If $\mathbf{e}' \neq \mathbf{e}(\sigma, g)$, she rejects the stream and quits.

4. PERFORMANCE

4.1. Security

The present scheme provides a proof of the multiple ESs of an MPEG-4 stream with an improved aggregated signature scheme such that the client can detect the following modifications:

1. one or more elementary streams are tampered or forged no matter how many old ESs are available to the attacker.
2. one or more signatures on the corresponding ESs are tampered or forged no matter how many "old" signatures are available to the attacker.
3. an author denies what he signed, instead he admits he signed another ES.

The general signature schemes (e.g., [13]) deal with the first two modifications, while the third one which binds an author with his ES is considered in aggregated signature only. We argue that the third one is also important. For example, given two authors P_1 and P_2, who sign two ESs m_1 and m_2 respectively. After the MPEG-4 stream including m_1 and m_2 is sent, the author P_1 is able to claim that he signs m_2 because m_2 is more valuable than m_1. Thanks to Eq.(1), the above vulnerability does not exist in our improved scheme because the public key v_1 is concatenated with the message m_1 with the assumption of one-way function..

4.2. Cost

Table 1 lists the comparison results between the aggregated signature and the individual signature given that there are n authors. In the Table, t_M denotes the computational cost of a modular multiplication, t_B denotes the operation cost of a bilinear mapping and $|\sigma|$ denotes the size of an individual signature in bits. In the Table, the computational cost of individual signature is fixed and hence ignored in the comparison.

Table 1. Comparison of individual (bilinear mapping based) signature scheme and aggregated signature scheme.

	Individual Signature	Present
Signing time	0	t_M
Verifying time	$(2n)t_B$	$(n-1)t_M + (n+1)t_B$
Overhead	$n\|\sigma\|$	$\|\sigma\|$

From Table 1, we know that our scheme requires one more modular multiplication, while for verification,

- the present scheme requires additional $(n-1)$ modular multiplication operations, but saves $(n-1)$ bilinear mapping operations. Since a bilinear mapping is much more expensive than a modular multiplication, the total verifying time is reduced greatly.
- the communication overhead of the present method is constant (one signature), whereas that of the individual signature based scheme is linear to the number of clients.

Experiment results on the BGLS signature scheme with 512-bit moduli were obtained in [18] using a P3-977Mhz Linux machine with the OpenSSL library for computing the individual operations. From the experiment results in [18], we can derive that $t_M = 0.12ms$ and $t_B = 31ms$, thus $(2n)t_B \gg (n-1)t_M + (n+1)t_B$. Therefore, the present authentic scheme outperforms the individual signature based scheme in terms of computational cost and communication overhead.

5. CONCLUSION

Aggregated signature scheme enables to combine an arbitrary number of signatures into only one. In this paper, we enhance the aggregated signature scheme [15] so as to bind the author with his ES with negligible cost. Additionally, we employ the improved scheme into MPEG-4 stream which includes a multiple of Elementary Streams so as to provide authentic distribution. The proposed scheme is efficient in terms of the computational cost and communication overhead since the client merely deals with only one signature for an MPEG-4 stream.

6. REFERENCES

[1] ISO/IEC 14496-1:2001(E), Information technology - Coding of audio-visual objects - Part 1: System, ISO/IEC JTC 1/SC 29/WG 11 N3850, 2000-10-19

[2] Fernando Pereira, Touradj Ebrahimi (ed.), The MPEG-4 Book, ISBN: 0130616214, Pearson Education, 2002

[3] ISO/IEC 14496-1:2001/FDAM 3:2003(E), Information technology - Coding of audio-visual objects - Part 1: Systems, AMENDMENT 3: Intellectual Property Management and Protection (IPMP) extensions, ISO/IEC JTC 1/SC 29/WG 11, 2002-12-4

[4] James King and Panos Kudumakis, "MPEG-4 IPMP Extension," DRM 2001, LNCS 2320, pp.126-140, 2002

[5] ISO/IEC 14496-13:2004(E), Information technology - Coding of audio-visual objects - Part 13: Intellectual Property Management and Protection (IPMP), extensions, SC 29/WG 11 N 5284, 2004-05-21

[6] T. Senoh, T. Ueno, T. Kogure, Shengmei Shen, Nfing Ji, Jing Liu, Zhongyang Huang, C. A. Schultz, "DRM renewability & interoperability," First IEEE Consumer Communications and Networking Conference (CCNC), pp.424-429, 2004

[7] MIRADOR: MPEG 4 Intellectual Property Rights by Adducing and Ordering, `http://www.cordis.lu/infowin/acts/analysys/products/thematic/mpeg4/mirador/mirador.htm`

[8] Jiangtao Wen, M. Severa, Wenjun Zeng, M.H. Luttrell, Weiyin Jin, "A format-compliant configurable encryption framework for access control of video," IEEE Trans. on Circuits and Systems for Video Technology, 12(6):545-557, 2002

[9] J. Lacy, N. Rump, T. Shamoon, and P. Kudumakis, "MPEG-4 Intellectual Property Management & Protection," 17th Conf. Audio Engineering Society, 1999.

[10] Kwang Yong Kim, JinWoo Hong, "MPEG4 IPMP authoring system for protection of object based contents," The 6th International Conference on Advanced Communication Technology, Vol.1, pp.499 - 503, 2004

[11] R. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, 1992

[12] National Institue of Standards and Technology, "Secure Hash Standard (SHS)", FIPS Publication 180-1, 1995.

[13] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, 21(2):120-126, 1978.

[14] National Institue of Standards and Technology, "Proposed Federal Information Processing Standard for Digital Signature Standard (DSS)," Federal Register, Vol. 56, No. 169, pp. 42980-42982, 1991.

[15] D.Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps", EUROCRYPT, Lecture Notes in Computer Science 2656, pp.416-432, 2003

[16] Igor E. Shparlinski, *Finite Fields: Theory and Computation*, pp.215-239, Kluwer Academic Publishers, ISBN 0-7923-5662-4, 1999.

[17] Florian Hess, "Efficient Identity based Signature Schemes based on Pairings," Selected Areas in Cryptography 2002, LNCS 2595, pp.310-324, 2003

[18] E. Mykletun, M. Narasimha, G. Tsudik, "Authentication and integrity in outsourced databases", Network and Distributed System Security Symposium(NDSS), 2004