2-2006

Scalable authentication of MPEG-4 streams

Yongdong WU

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1109/TMM.2005.861283

Scalable Authentication of MPEG-4 Streams

Yongdong Wu and Robert H. Deng

***Abstract*—This paper presents three scalable and efficient schemes for authenticating MPEG-4 streams: the Flat Authentication Scheme, the Progressive Authentication Scheme, and the Hierarchical Authentication Scheme. All the schemes allow authentication of MPEG-4 streams over lossy networks by integrating seamlessly digital signatures and erasure correction coding with MPEG-4's fine granular scalability. A prominent feature of our schemes is their "sign once, verify many ways" property, i.e., they generate only one digital signature per compressed MPEG-4 object group, but allow clients to verify the authenticity of any down-scaled version of the original signed object group.**

***Index Terms*—Authentication, digital signature, erasure correction coding, fine granular scalability, Merkle hash tree, MPEG-4.**

I. Introduction

MPEG-4 [1], [2] is a state-of-the-art multimedia processing technology that encompasses a wide range of tools and techniques. It not only deals with media compression but also covers media packaging and delivery, including delivery multimedia integration framework (DMIF) [3] and fine granular scalability (FGS) [4], as well as intellectual property management and protection (IPMP) [5].

DMIF is a generic multimedia platform for delivering synchronized and multiplexed media object streams. In DMIF, a large number of clients can access MPEG-4 media at any time and from anywhere, based on their preferences, communication channel characteristics, and device capabilities.

FGS is adopted by the MPEG-4 standard for efficient and flexible distribution of multimedia over heterogeneous wired and wireless networks [6]–[8]. FGS provides a down-scaling mechanism which adapts MPEG-4 media to network traffic and/or consumer device resource.

IPMP is a secure delivery framework which enables content providers to select and configure the most effective and appropriate tools for content protection. Currently, IPMP only provides provisions for elementary stream identification. It does not provide mechanisms for media stream authentication. However, authentication (including data integrity) of MPEG-4 stream is indispensable in certain applications, such as government, finance, health care and law. It is well known that using encryption without adequate integrity protection is vulnerable to active attacks [9] and it is believed that integrity service must be offered in any security-aware transmission [10]. Therefore, techniques for authenticating MPEG-4 streams are crucial for MPEG-4 to be used in security sensitive applications.

This paper proposes three authentication schemes for MPEG-4: the flat authentication scheme (FAS), the progressive authentication scheme (PAS), and the hierarchical authentication scheme (HAS). All the three schemes perform the following operations: 1) video objects are assembled into object groups and each object group is encapsulated into a packet group; 2) one digital signature per packet group is produced on the hash values of packets in the packet group; 3) erasure correction codewords are generated from the signature and hash values of packets which are then amortized into the packets; 4) a proxy in the distribution network down-scales an object group based on the priorities of the objects and produces patches which are piggybacked onto the packets; and 5) clients verify the authenticity of the source of the received packets. In this framework, the critical issue is how to manage packet authentication. FAS manages the authentication issue using an incremental hash function, PAS does it based on a progressive hash function, and HAS deals with the problem employing a hash tree. Since the proposed schemes seamlessly integrate stream authentication with erasure correction coding, as well as with an unequal protection technique based on object priorities, they can tolerate packet loss while still achieve authentication with low overhead. Additionally, the schemes exploit the FGS property of MPEG-4 so that they are able to authenticate down-scaled object groups with only one digital signature per object group. That is, they allow proxies to down-scale object groups such that clients can verify the authenticity of the source of any down-scaled streams.

A. Related Works

To cater for the FGS property of MPEG-4, Achir *et al.* [11] proposed two MPEG-4 delivery algorithms which are able to adapt to time-varying multicast networks. The first algorithm, namely Layer-based Priority, maintains the best possible quality for each video object. The second algorithm, called Object-based Priority, guarantees the best possible quality for the most important object but coarsely degrades the less important objects of a video stream. Yuan *et al.* [12] provided a method for scalable encryption of MPEG-4 streams. Venkatramani *et al.* [13] designed ARMS so as to provide secure end-to-end MPEG-4 based streaming. However, the previous efforts did not take stream authentication into consideration.

A straightforward stream authentication technique is to append a digital signature to each packet. This naïve solution not only introduces high computational overhead (e.g., it requires 4.65 ms to generate a 1024-bit RSA signature with a 2.1-GHz Pentium 4 processor [14]), but also large communication overhead. In addition, FGS makes stream authentication even more

Manuscript received November 23, 2003; revised March 28, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ton Kalker.

Y. Wu is with the Information Security Department, Institute for Infocomm Research, Singapore (e-mail: wydong@i2r.a-star.edu.sg).

R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore (e-mail: robertdeng@smu.edu.sg).

Digital Object Identifier 10.1109/TMM.2005.861283

challenging because it allows MPEG-4 streams to be modified intentionally and legally. We note that letting intermediate servers to sign down-scaled streams is impractical and insecure.

Miner and Staddon [15] proposed to authenticate digital streams over a lossy network based on a packet-dependent graph. Their authentication scheme tolerates random packet loss given that each packet is independently lost with the same probability. However, the scheme is not applicable for scalable stream authentication.

Park *et al.* [16] constructed an authentication scheme SAIDA by encoding the hash values of packets and the signature on a packet group with an erasure correction coding algorithm. Their method amortizes codewords for packet hash values and packet group signature among all the packets so as to reduce space overhead and increase tolerance of packet loss. Once sufficient number of packets are received, the receiver starts to recover the signature and check authenticity of the packets. Pannetrat *et al.* [17] improved SAIDA by constructing a systematic erasure code to reduce the packet overhead. We extend Pannetrat's approach to provide flexible authentication of MPEG-4 streams.

Wu and Deng [18] also extended the scheme in [16] but focused on the semantical authenticity of Motion JPEG2000 streams. They did not consider the effect of down-scaling operations which is one of the main concerns in this paper. A number of other cryptographic based stream authentication schemes have been studied, the reader is directed to [19]–[26] for technical details.

A watermark based stream authentication scheme was proposed by Lin and Chang [27]. The scheme extracts the invariable features to reject malicious tampering but accept nonmalicious modifications. However, it is in general hard to measure the distortion formally because the criteria itself is subjective. Furthermore, this scheme is vulnerable to known-plaintext attack.

B. Outline

The rest of this paper is organized as follows. Section II introduces preliminaries. The underlying framework of the proposed stream authentication schemes is addressed in Section III. Sections IV–VI elaborate on the details of our three authentication schemes, respectively. Section VII contains the security and performance analysis. Section VIII concludes the paper.

II. Preliminaries

To make the paper self contained, this section introduces the basic concepts of one-way hash function, digital signature, Merkle hash tree, erasure correction coding (ECC), and MPEG-4.

A. 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 $\mathcal{H}(\cdot)$, is a hash function that 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 believe-to-be one-way hash functions, such as SHA [28].

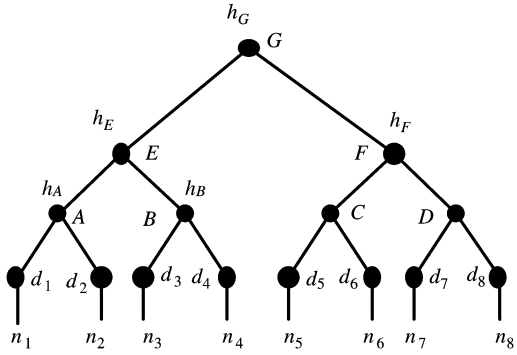

Fig. 1. Example Merkle hash tree where $d_i = \mathcal{H}(n_i \parallel i), i = 1, 2, \ldots, 8$.

B. Digital Signature

A digital signature algorithm is a cryptographic tool for generating nonrepudiation evidence, authenticating the integrity as well as the origin of the signed message. In a digital signature algorithm, a signer keeps a private key K_s and publishes the corresponding public key K_e. The private key is used by the signer to generate a digital signature σ on a message m based on a signature generation function $\sigma = Sign(K_s, m)$, and the public key K_e is used by anyone to verify the signature σ on the message m based on a verification function $Veri(m, \sigma, K_e)$. The digital signature algorithms widely used in practice include RSA [29] and DSA [30].

C. The Merkle Hash Tree

The Merkle hash tree finds many applications, e.g., [31]–[35]. We illustrate the construction and application of the Merkle hash tree with a simple example. The reader is referred to [36] for detailed descriptions. To authenticate data values $n_1, n_2, \ldots, n_w$, the data source constructs the Merkle hash tree as depicted in Fig. 1 assuming that $w = 8$. Each node in the tree is assigned a value. The values of the 8 leaf nodes are the hash values, $d_i = \mathcal{H}(n_i \parallel i)$, $i = 1, 2, \ldots, 8$, of the data values under a one-way hash function $\mathcal{H}(\cdot)$, where "||" denotes concatenation. The value of each internal node of the tree is derived from its child nodes. For example, the values of node A and node B are $h_A = \mathcal{H}(d_1 \parallel d_2)$ and $h_B = \mathcal{H}(d_3 \parallel d_4)$, respectively. The value of node E is $h_E = \mathcal{H}(h_A \parallel h_B)$. The data source completes the construction of the tree recursively from the leaf nodes to the root node.

The value of the root node G is $h_G = \mathcal{H}(h_E \parallel h_F)$ which is used to commit to the entire tree; it can be used to authenticate any subset of the data values $(n_1, \ldots, n_8)$, in conjunction with a small amount of auxiliary information. For example, a client, who is assumed to have the authentic root value h_G, requests for n_3 and requires the authentication of the received n_3. Besides n_3, the source sends the auxiliary information d_4, h_A and h_F to the client. The client can then check the authenticity of the received n_3 as follows. The client first computes d_3 and $\mathcal{H}(\mathcal{H}(h_A \parallel \mathcal{H}(d_3 \parallel d_4)) \parallel h_F)$ and then checks if the latter is the same as the root value h_G. If this check is positive, the client accepts n_3. In general, to authenticate the data value n_i, the auxiliary information include the values of all the sibling nodes of those nodes on the path from the leaf node d_i to the root.

D. Erasure Correction Coding

Generally, network errors include packet loss and packet data errors. However, in MPEG-4 stream applications, only packet loss is considered because packet data errors are detected with packet checksum and thus packets with errors are dropped, which results in lost packets at the end users. Furthermore, since packet identification can be used to locate missing packets, systematic ECC is a good means for error resilience in our context. An (n, k) ECC scheme consists of an encoder module and a decoder module. The encoder module accepts a message of k symbols over the finite field $GF(2^w)$ and outputs a codeword of n symbols. Assuming that the minimum Hamming distance of the code is d. Then the decoder module is able to recover the original k message symbols given any $n - d + 1$ symbols in a received codeword [37]. In (n, k) ECC, an appropriate $n \times k$ generator matrix $\mathbf{G}$ is selected such that any k rows of $\mathbf{G}$ form a nonregular matrix. A systematic (n, k) ECC requires the first k rows of $\mathbf{G}$ be an identity matrix, i.e.,

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & 1 \\ a_{k+1,1} & a_{k+1,2} & \dots & a_{k+1,k} \\ \vdots & \ddots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,k} \end{pmatrix}$$

where $a_{i,j} \in GF(2^w)$[37], [38]. Denote a message as $m = (m_1, m_2, \dots, m_k)^T$, $m_i \in GF(2^w)$ for all i, the corresponding codeword is given by $C = \mathbf{G}m = (m_1, m_2, \dots, m_k, c_1, c_2, \dots, c_{n-k})^T$, where c_i, $i = 1, 2, \dots, n-k$ are parity symbols. In this paper, we assume that the (n, k) ECC scheme is maximum distance separable, i.e., its minimum Hamming distance $d = n - k + 1$ [37]. Then given any k or more symbols in a received codeword, the decoder outputs the original message of k symbols.

E. Syntactic Structure of MPEG-4

According to [1], [39], an MPEG-4 presentation is divided into sessions including units of aural, visual, or audiovisual content, called media objects. As shown in Fig. 2, a video sequence or group includes a series of video objects (VOs). Each VO is encoded in one or more video object layers (VOLs). Each layer includes information corresponding to a given level of temporal and spatial resolution, so that scalable transmission and storage are possible. Each VOL contains a sequence of two-dimensional (2-D) representations of arbitrary shapes at different time intervals that is referred to as a video object plane (VOP). Video object planes are divided further into macroblocks (MBs) of size 16×16. Each macroblock is encoded into six blocks $B_1, B_2, \dots, B_6$ of size 8×8 when a 4:2:0 format is applied.

In a MPEG-4 stream, video objects, such as foreground objects and background objects, may have different priorities, indicated as *visual_object_priority* taking values 1–7 from the lowest to the highest priority. In the MPEG-4 syntax, each object layer has *visual_object_layer_priority* to represent the importance of different layers. The layer with the highest priority, called the base layer, contains data representing the most important features of the video sequence, while additional layers, called enhancement layers, assigned with progressively lower priorities, contain data that further refines the quality of the base layer. The source generates a flow for each layer and assigns to it a unique discarding priority.

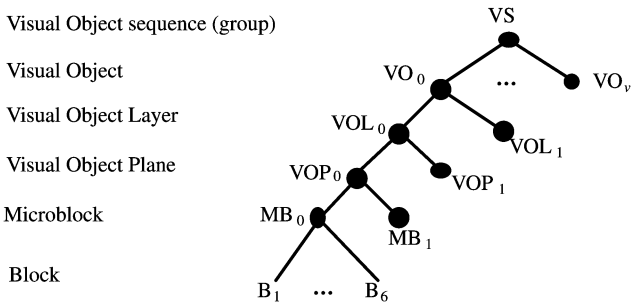

Fig. 2. Syntactic structure of an example MPEG-4 visual stream. This stream includes v objects, each object has two VOLs, each VOL includes two VOPs and each VOP is segmented into two macroblocks.

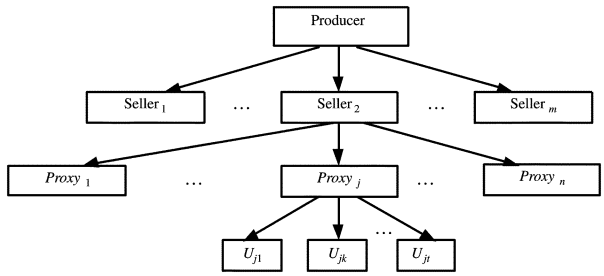

Fig. 3. Content distribution framework where a transmission path may include many proxies.

III. Framework of Proposed Schemes

A. Problem Definition

Fig. 3 illustrates the generic parties involved in the content dissemination process. A producer generates a protected MPEG-4 stream and disseminates it to the sellers (e.g., edge servers of a content distribution network); each seller passes the content to proxies in one or more transmission stages. This process continues till the stream arrives at the clients. In this framework, only the producer is assumed to be trustworthy. For instance, only the certified public key of a digital video camera is available and trusted in a real-time monitoring application.

In packet lossy networks such as Internet, packet loss increases the difficulty of authenticating streams. Packet loss comes from three main sources:

- a proxy discards unimportant content intentionally so as to meet the network and client device requirements;
- a router discards packets due to network limitation;
- a receiver discards packets which fail checksum verifications. Although error resilience methods (e.g., [40]–[42]) for MPEG-4 video streams over the Internet have been proposed for video quality, packet loss still happens from time to time.

The objective of a stream authentication scheme is, with high probability and using as fewer signatures as possible, to authenticate packets received over a lossy network. Accordingly,

a stream authentication scheme for MPEG-4 should answer the following questions: How to reduce the computational and communication cost? How to increase the probability of successful authentication in case of packet loss? How to manage data removal at proxies so as to allow successful authentication?

B. Overview of the Proposed Schemes

Following the approaches in [19]–[26], our schemes divide a stream into groups of packets (e. g., grouping packets corresponding to a certain video duration). To facilitate down-scaling operation at proxies, a packet group includes data of one or more VOs. Because groups are processed independently, we focus on one packet group G in the following. In our schemes, the producer signs on the hash value of the group, instead of on each packet separately. The group hash is generated such that recipients are able to verify the source of a down-scaled stream. The time to run signing and verifying algorithms, as well as the authentication data, are amortized over many packets. Consequently, this approach keeps the authentication overhead low.

To achieve packet loss resilience with small communication overhead, the objects, and/or layers are given unequal protection levels [43]. That is, the higher priority of an object/layer is, the more parity symbols are used to protect it. To ensure that the highest priority layer (i. e., the base layer) be always verifiable, the signature is protected at the same level as the highest priority layer.

Because FGS is used in MPEG-4 streaming, a proxy discards data layers from the lowest priority layer to higher priority layers until the resource requirement is met. This down-scaling strategy is different from the packet dropping method adopted in [44]. Because the down-scaled stream can tolerate the same number of packet losses as the original stream, the error-resilience capability is not decreased. Thus, a client is able to verify authenticity of the packet origin even if the stream is down-scaled. Of course this assumes that proxies are aware of the presence of the authentication mechanisms.

Briefly, the proposed authentication schemes work as follows (refer to Fig. 4).

- The producer encodes video objects according to the MPEG-4 standard.
- The producer prepares packets for an object group based on the priorities of video objects and layers.
- The producer generates authentication data including packet hashes and a signature. The authentication data is piggybacked/appended onto the packets. The protected stream is then disseminated over the network.
- To meet the requirement of the network bandwidth or the client device capability, proxies may down-scale the stream in such a way that the authenticity of the down-scaled stream remains verifiable at clients.
- A client recovers the signature and packet hashes, and verifies the received stream. Only when a packet is authentic, the client will accept and decode it.

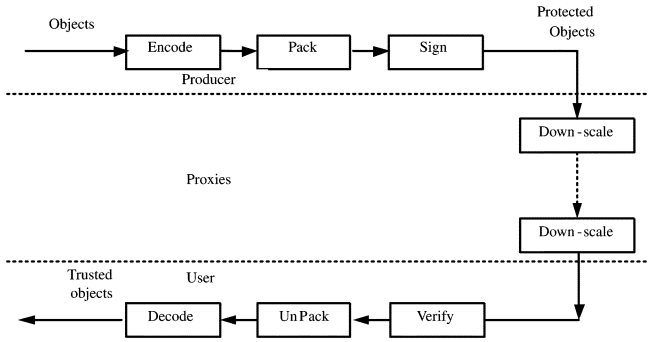

Fig. 4. Process of stream authentication.

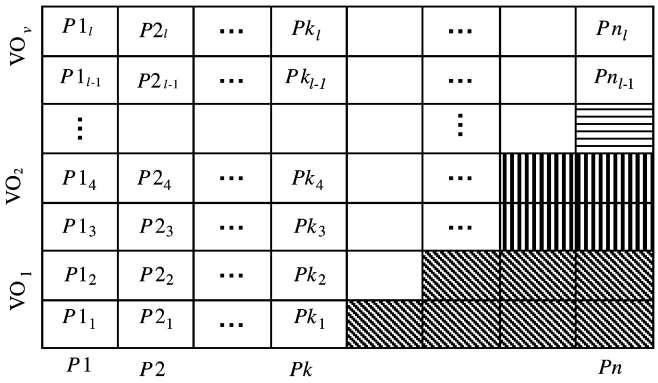

Fig. 5. Packaging of an object group where data units in the same column forms a packet, while data units in the same row form a layer.

IV. Flat Authentication Scheme

In the flat authentication scheme (FAS), a proxy can discard any number of VOs, VOLs, VOPs, macroblocks, blocks or their arbitrary combinations, but the authenticity proof is still valid and the authentication overhead is kept constant. In Sections IV-A–E, we describe FAS according to the framework shown in Section III.

A. Packaging an Object Group

Layers in an object group G are arranged based on a predefined style [43]. Fig. 5 illustrates the arrangement of a group of v objects. Objects are encapsulated into n packets (i.e., columns) $P1, P2, \ldots, Pn$, where packet Pi consists of data units Pi_j, $j = 1, 2, \ldots, l$, with l being the number of layers. In Fig. 5 data are protected unequally with the shaded areas representing parity units. That is, important layers consist of more parity units and less important layers have less parity units.

B. Generating Signature on an Object Group

To achieve maximum flexibility, we make use of a communicative function $f(\cdot)$ to generate packet hashes. Formally

$$f(x_1, \ldots, x_i, \ldots, x_j, \ldots, x_l) = f(x_1, \ldots, x_j, \ldots, x_i, \ldots, x_l)$$

for any $i, j = 1, 2, \ldots, l$. We select the incremental hash algorithm **AdHASH** in [45] due to its efficiency. Thus, the incremental hash of packet P_i is

$$h_i = HLi_1 \oplus HLi_2 \oplus \cdots \oplus HLi_l \quad (1)$$

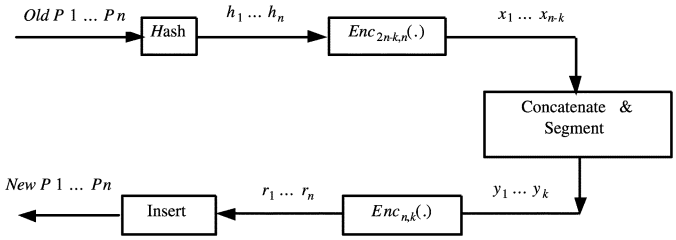

Fig. 6. Process to generate integrity-protected packets.

where HLi_j is the hash value of data unit Pi_j in packet Pi

$$HLi_j = \mathcal{H}(Pi_j \parallel j), \quad j = 1, 2 \ldots, l. \tag{2}$$

We define the packet hash of P_i as

$$g_i = \mathcal{H}(h_i \parallel i) \tag{3}$$

and the hash value of the object group G as

$$h_G = \mathcal{H}(g_1 \parallel g_2 \parallel \cdots \parallel g_n \parallel G_{\mathrm{ID}} \parallel S_{\mathrm{ID}}) \tag{4}$$

where G_{ID} is the identification of the group G and S_{ID} is the identification of the stream (e.g., video number). Note that the packet index i and and the group identification G_{ID} are inserted into (3) and (4), respectively, so as to defeat a counterfeiting attack [46]. Finally, the producer signs on the group hash h_G using its private key K_s to obtain the signature $\sigma = Sign(K_s, h_G)$.

C. Encoding and Encapsulating

After generating the signature, ECC encoders are employed to encode the signature and packet hashes so as to tolerate packet losses (see Fig. 6).

1) Generate a codeword $X = (h_1, h_2, \ldots, h_n, x_1 \ldots, x_{n-k})^T = Enc_{2n-k,n}(h_1, h_2, \ldots, h_n)$, where $Enc_{2n-k,n}(\cdot)$ is a systematic ECC algorithm with symbols over field $GF(2^{w_1})$, n is the number of packets in a group, and k is the minimum number of expected received packets.
2) Divide the concatenation $(x_1 \parallel x_2 \parallel \cdots \parallel x_{n-k})$ into k symbols $y_i \in GF(2^{w_2})$, $i = 1, 2, \ldots, k$. With the ECC algorithm, a codeword $C_r = Enc_{n,k}(y_1, y_2, \ldots, y_k)$ is produced. Denote the n symbols in the codeword C_r as integrity units $r_1, r_2, \ldots, r_n$.
3) Similarly, divide the signature σ into k symbols σ_i of the same size, $i = 1, 2, \ldots, k$. $\sigma_i \in GF(2^{w_3})$. Then encode the signature to produce a signature codeword $C_s = Enc_{n,k}(\sigma_1, \sigma_2, \ldots, \sigma_k)$. Denote the n symbols in the codeword C_s as signature units $s_1, s_2, \ldots, s_n$.
4) Append integrity unit r_i and signature unit s_i to the original packet Pi, for all $i = 1, 2, \ldots, n$. That is, the new packet Pi now consists of r_i, s_i and $Pi_1, Pi_2, \ldots, Pi_l$. Fig. 7 illustrates the packet structure of the protected object group.

D. Down-Scaling Objects

After generating the protected object group, the producer sends protected packets $P1, P2, \ldots, Pn$ over the content distribution network. As mentioned previously, a proxy may need to customize the object stream so as to adapt it to the available network bandwidth. In this case, the proxy may discard unimportant layers. How to select the down-scaling layers is beyond the scope of this paper. To prove the source of the down-scaled object group to clients, for each packet, the proxy sums the hashes of the discarded data units, and inserts the summation into the packet. Specifically, if layers $t+1, t+2, \ldots, l$ are discarded by a proxy, a patch value

$$e_i = HLi_{t+1} \oplus HLi_{t+2} \oplus \cdots \oplus HLi_l \tag{5}$$

is inserted into packet Pi, for all $i = 1, 2, \ldots, n$. Albeit (5) indicates the contiguous layers being removed, FAS is applicable to random layer removal given that the receiver is acknowledged with the identities of the removed layers. If more than one proxy down-scale the objects, the new patch will be added to the old one, and the overhead will be kept constant. Fig. 8 demonstrates the down-scaled protected group due to removable of all objects but VO_1 and VO_2 (four layers) by a proxy.

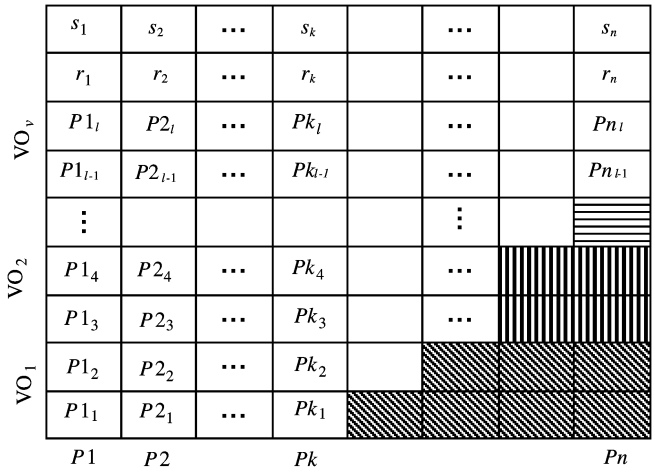

Fig. 7. Packet structure of the protected object group.

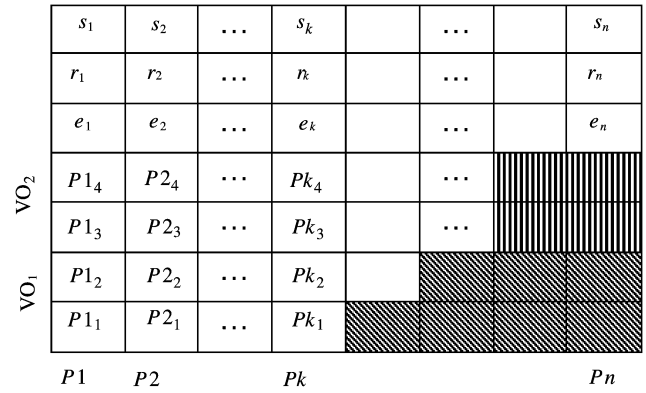

Fig. 8. Down-scaled object group where the third row contains the patch values.

E. Verifying Packets

The verification process basically reverses the generation process for a protected object group. Based on the erasure coding, at least k packets of a group should be received in order to recover the signature and packet hash values. Without loss of generality, suppose the first k packets $P1, P2, \ldots, Pk$ are received successfully. With respect to Fig. 9, the client retrieves

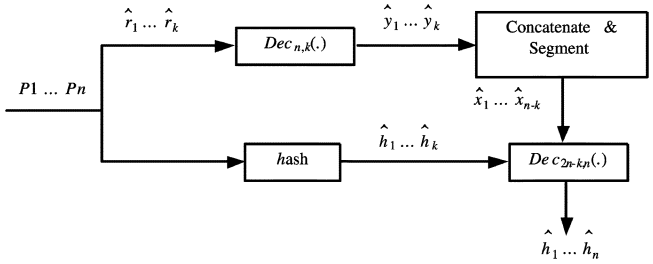

Fig. 9. Process of recovering the packet hashes.

the integrity units $\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_k$ from the received packets and using the decoding algorithm $Dec_{n,k}(\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_k)$ to recover $\hat{y}_i$ $(i = 1, 2, \ldots, k)$, or $\hat{x}_i$ $(i = 1, 2, \ldots, n-k)$. From the received packets, the client calculates

$$\hat{h}_i = \left(\bigoplus_{j=1}^{t} HLi_j \right) \oplus e_i, \quad i = 1, 2, \ldots, k$$

and then using $Dec_{2n-k,n}(\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_k, \hat{x}_1, \ldots, \hat{x}_{n-k})$ to obtain the hash value $\hat{g}_i$ corresponding to packet Pi, $i = 1, 2, \ldots, n$. The reconstructed hash for the group G is

$$\hat{h}_G = \mathcal{H}(\hat{g}_1 \parallel \hat{g}_2 \parallel \cdots \parallel \hat{g}_n \parallel G_{\mathrm{id}} \parallel S_{\mathrm{id}}).$$

Meanwhile, signature units $s_1, s_2, \ldots, s_k$ are retrieved too. With the decoder $Dec_{n,k}(s_1, s_2, \ldots, s_k)$, the signature $\hat{\sigma}$ for the group is reconstructed. Therefore, the signature can be verified against $\hat{h}_G$ with algorithm $Veri(\hat{h}_G, \hat{\sigma}, K_e)$, where K_e is the authentic public key of the producer. If $\hat{h}_G$ does not match with $\hat{\sigma}$, the object group is considered forged and discarded.

V. Progressive Authentication Scheme

FAS is flexible in that it allows a proxy to discard any layers while still ensures that the authenticity of the dow-scaled object group is verifiable at the clients. However, FAS is only viable when an adversary eavesdrops the network, drops packets, or modify the packets randomly, as we will discuss in Section VII. To foil a strong adversary who is able to control the network, **AcHASH** [47][1], instead of **AdHASH**, can be employed for generating packet hashes. Although **AcHASH** is secure, it is however computationally very expensive.

In this section, we propose the PAS which is built upon a progressive hash function based on the layered structure of MPEG-4. Similar to FAS, PAS encapsulates objects into packets as shown in Fig. 5. Recall that the layout in Fig. 5 is based on the priorities of layers, i.e., layer i has higher priority than layer $i+1$. The layer containing Pi_1 is the most important layer, while the layer containing Pi_l is the least important layer. Because the layers are totally ordered, it makes sense to discard layer $i+1$ ahead of layer i if a down-scaling operation is necessary.

[1]**AcHASH** is a one-way accumulator defined as $f(x_1, x_2, \ldots, x_n) = a^{x_1 x_2 \cdots x_n} \bmod N$, where a is a constant and N is the product of two large primes.

A. Generating Signature on an Object Group

We define the progressive hash of the packet P_i as

$$g_i = \mathcal{H}(\mathcal{H}(Pi_1 \parallel \mathcal{H}(Pi_2 \parallel \mathcal{H}(\ldots \parallel \mathcal{H}(Pi_l)))) \parallel i) \quad (6)$$

where the priority of data unit Pi_j is higher than the priority of data unit $Pi_{j+1}, j = 1, 2, \ldots, l-1$. The steps for signature generating and encapsulating are the same as those in Section IV-C.

B. Down-Scaling Objects

After generating the protected object group, the producer sends packets $P1, P2, \ldots, Pn$ over the content distribution network. When several layers are removed at a proxy, the hash values of the discarded layers are inserted into the packets. Specifically, if layers $t+1, t+2, \ldots, l$ are removed, the patch value for packet Pi is

$$e_i = \mathcal{H}(Pi_{t+1} \parallel \mathcal{H}(Pi_{t+2} \parallel \mathcal{H}(\ldots \parallel \mathcal{H}(Pi_l))))$$

for all $i = 1, 2, \ldots, n$. The proxy inserts the patch to the corresponding packet P_i. The data structure of the protected group generated with PAS is the same as that generated with FAS except that the patch values are different (see Fig. 8). If more than one proxy down-scale the objects, the new patch will replace the old one, and the payload will not increase.

C. Verifying Packets

To verify the authenticity of a down-scaled object group, a client recovers the signature and hash values $\hat{g}_i$ of all packets. This process is the same as that in FAS except that the process to recompute the hash value of a packet is different. If layers $t+1, t+2, \ldots, l$ are removed, the hash value for packet Pi can be recovered from

$$\tilde{g}_i = \mathcal{H}(\mathcal{H}(Pi_1 \parallel \mathcal{H}(Pi_2 \parallel \cdots) \parallel e_i) \parallel i) \quad (7)$$

where e_i is the patch value extracted from the packet P_i. The rest of the verification process is the same as in Section IV-E.

VI. Hierarchical Authentication Scheme

FAS is flexible but subject to attacks by malicious proxies. On the other hand, PAS is secure against malicious proxies (see Section VII) but requires that the layers be totally ordered. This total ordering requirement may not be met for certain applications, e.g., applications where some objects have the same priority. The HAS proposed in this section combines the advantages of FAS and PAS so as to provide a secure and flexible solution. HAS packages data units as FAS, but uses an object tree other than FAS's linear structure to manage packet integrity. In the general object tree structure shown in Fig. 2, the first layer corresponds to VO, the following layers to VOL, VOP, MB, and the leaf layer to blocks. HAS uses a tree which arranges VOLs/VOs based on their priority levels. Each object VO corresponds to an object subtree and all the object subtrees form the complete object group tree. Subtrees of objects with lower priority levels are placed as child subtrees of objects with higher priority levels. Within an object subtree, each object layer VOL corresponds to an object layer subtree. Subtrees of object layers

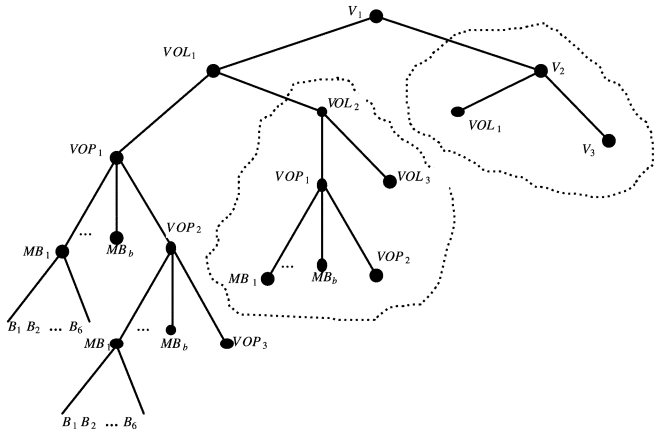

Fig. 10. Tree structure for the layers based on priorities. The dashed frames are covering-subtrees, which may be removed by a proxy.

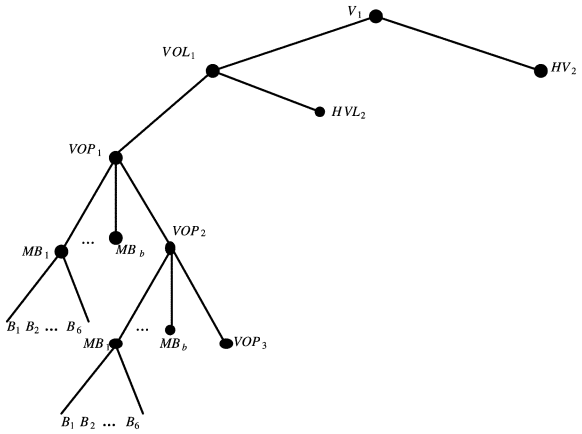

Fig. 11. Down-scaled object group. Only VOL_1 of object V_1 is not removed. HV_2 and HVL_2 are hash values of covering-subtree for the discarded layers.

with lower priority layers are arranged as child subtrees of object layers of lower priority levels. An example object group hash tree is depicted in Fig. 10. Note that there might be specific variations of the tree structure for specific applications. For example, if no cut operation is permitted, nodes correspond to macroblocks will be the leaf nodes.

A. Generating Signature on an Object Group

After defining the object group tree structure, the nodes in the tree will be assigned values recursively as the Merkle hash tree. First, leaf node values are generated with formula (2). Then, the value N_i of a nonleaf node is the hash value of its child node values

$$N_i = \mathcal{H}(D_1 \parallel D_2 \parallel \cdots \parallel D_c)$$

where c is the number of child nodes, and D_j is the value of the child node j.

For example, assume B_j's in Fig. 10 are within the same packets, then the node hash values are given by

$$\text{Block}: L1_j = \mathcal{H}(B_j \parallel j) \tag{8}$$

$$\text{MB}: L2_j = \mathcal{H}((L1_1 \parallel \cdots \parallel L1_6)) \tag{9}$$

$$\text{VOP}: L3_j = \mathcal{H}((L2_1 \parallel \cdots \parallel L2_b) \parallel L3_{j+1}) \tag{10}$$

$$\text{VOL}: L4_j = \mathcal{H}(L3_1 \parallel L4_{j+1}) \tag{11}$$

$$\text{VO}: L5_j = \mathcal{H}(L4_1 \parallel L5_{j+1}) \tag{12}$$

and the packet hash is given by

$$g_i = \mathcal{H}(L5_1 \parallel i) \tag{13}$$

where (8) computes the hash of each block, (9) calculates the hash of each macroblock and (11) and (12) calculate, respectively, the hashes of each object layer and object recursively.

Finally, the object group hash is

$$h_G = \mathcal{H}(g_1 \parallel g_2 \parallel \cdots \parallel g_n \parallel G_{\mathrm{id}} \parallel S_{\mathrm{id}}).$$

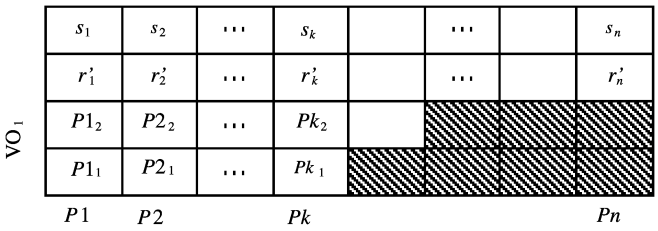

Fig. 12. Packet structure of a down-scaled object group. Only the most important layer of the most important object is kept.

The producer now signs the group hash h_G using its private key K_s to obtain the signature $\sigma = Sign(K_s, h_G)$. The process of encapsulating signature and packet integrity is the same as those in Section IV-C except that the integrity unit r_i is generated from the hash of the leaves in Fig. 10 instead of the data unit. Therefore, it is clean to encapsulate leaf values in packets.

B. Down-Scaling Objects

Define a covering-subtree as a subtree which includes a set of leaf nodes and only those leaf nodes. If a proxy removes layers, it will replace the layers with the hash value of the covering sub-tree in order for clients to verify authenticity. Referring to Fig. 11, all the objects but object V_1 are removed from all the packets, and all the VOLs of the object V_1 but the base layer VOL_1 are removed. In the corresponding hash tree, their hash values are used to replace the sub-trees derived from the layer data. Fig. 12 illustrates the data structure of a down-scaled object group. HVL_2 and HV_2 are the hashes of the new leaves shown in Fig. 11. To prove the authenticity of the down-scaled groups, the proxy will re-encode the hashes of the new tree leaves so as to generate the new r'_i which replaces the old integrity unit r_i.

C. Verifying Packets

After recovering the signature and the hash values $\hat{g}_i$ of a packet, the client reconstructs the tree of the packet layers, e.g., the tree shown in Fig. 11. Then the client computes the hash value $\tilde{g}_i$ of the hash tree according to formulas (8)–(13). If some

layers are removed, the hash value of the covering-subtree is used as the value of the node. The rest of the verification process is similar to that in Section IV-E.

VII. Security and Performance

In this section, we study the performance of the proposed schemes in terms of authentication probability, security, and computational cost. FAS is efficient in both storage and transmission. It also allows proxies to down-scale MPEG-4 streams in a flexible manner, but is only suitable for applications with moderate security requirement. PAS provides strong security protection but it requires that all layers be totally ordered so that layers can be down-scaled/removed by proxies in a predefined way. HAS provides both high security and great flexibility.

A. Authentication Probability

To provide reasonably high probability of successful authentication, an authentication scheme must have high probability of correctly receiving and verifying the base layer. Thus, the number of parity units for base layer should not be less than that for the signature; meanwhile, if the signature codeword has more parity units than that of base layer, then the signature is over protected. Thus, the number of parity units for signature and base layer should be the same. Assume that the packet losses are independent, then the received packets can be verified with probability $\sum_{i=0}^{n-k}\binom{n}{i}p^i(1-p)^{n-i}$[2], where p is the packet loss probability. To provide a concrete evaluation of the authentication probabilities, we use the experiment results from [48, Table II] where the stationary packet loss probability $p = 0.0222$. The received packets will be verified to be authentic as long as the number of lost packets $X \le n-k$. Then the authentication probability is given by

$$P(X \le n-k) = \sum_{i=0}^{n-k}\binom{n}{i}p^i(1-p)^{n-i} = \sum_{i=0}^{n-k}\binom{n}{i}0.0222^i \times 0.9778^{n-i}.$$

Assuming a video rate of 25 frames/s and each frame is encapsulated in a packet, Fig. 13 illustrates the authentication probabilities for packet groups of size $n = 100$, 200 and 400 (i.e, 4 s, 8 s, and 16 s durations), respectively, given that the packet loss rate is $p = 0.0222$. Let $R = (n-k)/n$ be a measure of coding redundance. First, we observe that for a fixed number of parity packets $n-k$, the authentication probability decreases as the group size n increases. This is understandable since when $n-k$ is fixed, a larger n corresponds to a smaller coding redundance. The more interesting observation is that we can achieve high authentication probabilities with small coding redundances. For example, the authentication probability is above 0.98 when R is about 5% for $n = 100$ and 200.

B. Security

In the following, we assume that the underlying one-way hash function $\mathcal{H}(\cdot)$ and the producer's signature scheme are secure. Moreover, the producer's public key K_e is disseminated to clients in an authenticated manner.

[2]For easy estimation of authentication probability, assume that each packet encapsulates the same number of leaves in the HAS tree.

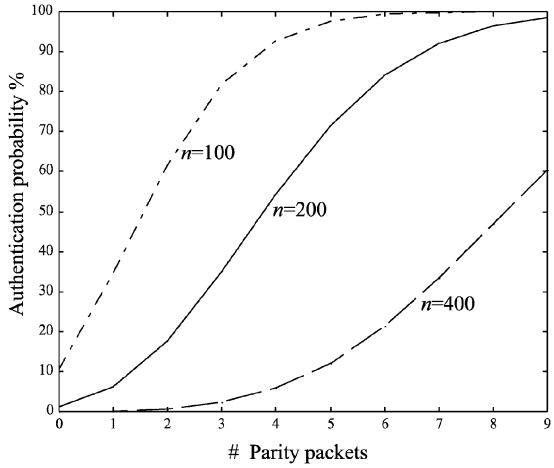

Fig. 13. Authentication probabilities versus number of parity packets.

1) Security of FAS: Packet hash values in FAS are computed using an incremental hash algorithm **AdHASH** [see (1)]. Since the hash algorithm is communicative, FAS allows a proxy to discard any layers and compute the corresponding patch values such that the down-scaled packet group remains verifiable at the clients. However, this same flexibility provides adversaries, such a malicious proxy, to forge data without being detected. To see this, let us consider packet Pi in Fig. 7 whose incremental hash h_i is given by (1). Assuming that layers $t+1, t+2, \ldots, l$ are discarded by a proxy. According to FAS, the proxy computes a patch value e_i given by (5). Obviously, we have $h_i = HLi_1 \oplus HLi_2 \oplus \cdots \oplus HLi_t \oplus e_i$. Now assuming that the proxy is malicious. The proxy modifies the undiscarded data units, say replacing Pi_1 with $\hat{P}i_1$. The proxy computes $\hat{H}Li_1 = \mathcal{H}(\hat{P}i_1)$ and the "patch value" $\hat{e}_i = \hat{H}Li_1 \oplus HLi_1 \oplus e_i$, where e_i is again given by (5). When this down-scaled packet is received by a client, the packet will pass verification since $\hat{H}Li_1 \oplus HLi_2 \oplus \cdots \oplus HLi_t \oplus \hat{e}_i = h_i$. It should be noted that to launch this attack, the attacker needs to get control of a proxy or edge server in the content distribution network. Therefore, FAS provides a moderate level of security and is suitable for scenarios where the proxies are trusted not to diverge from the correct protocol operation.

2) Security of PAS: PAS employs a progressive hash function in computing the packet hash value as shown in (6). Due to the recursive hash operations of the progressive hash function, the above attack to FAS by a malicious proxy is computationally impossible in PAS. Note that the malicious proxy attack is an "insider attack" and is the most powerful attack to our application scenario. Other attacks such as content removal, re-arrangement, and modification by third parties can by detected as long as the underlying hash function and the signature scheme are secure. Authentication of origin of the content also depends on the authenticated dissemination of the producer's public key to clients, a necessary requirement in any signature based authentication schemes. We note that to facilitate packet hash computation, and therefore packet verification, layers in PAS must be discarded in a pre-determined manner (e. g., from the lowest pri-

ority layer to the higher priority layers) during the down-scale operation at a proxy.

3) Security of HAS: The security of HAS depends on the security of the Merkle hash tree. The Merkle hash tree can prevent an adversary, who impersonates as the producer, from sending forged data to the client. With reference to Fig. 1, an adversary impersonating as the source can not sends a forged n'_3 to the client, because he cannot find $\mathcal{H}(n'_4)$, h'_A and h'_F such that $\mathcal{H}(\mathcal{H}(h'_A \parallel \mathcal{H}(\mathcal{H}(n'_3) \parallel \mathcal{H}(n'_4))) \parallel h'_F) = h_G$. Based on the same argument, the Merkle hash tree can also detect modifications made by any third party, including the proxies, as long as the producer's signature scheme and the underlying hash function are secure. More details about the security of the Merkle hash tree are described in [36].

C. Computational Cost

A producer is required to prepare protected streams, including encoding, packaging and signing. The task of a proxy is only to down-scale the objects and generate the packet patches. This lightweight task enables the proxy to process many simultaneous requirements from different clients. A client has to spend time in verifying the packets and decoding the objects. In the following, we study the cost related to security and ignore the cost due to object encoding/decoding.

Based on (2) and (3), (6) and (8)–(13), the number of hash operations per packet are $l+1$, $l+1$ and roughly $2l$ for FAS, PAS and HAS, respectively, at the producer side. The computational cost at a proxy is linear with the number of removed layers. The client will spend time to reconstruct the hash value of each packet. Loosely speaking, the total cost of the proxy and the client for computing the packet hashes is equal to that of the producer. The computational cost for signature verification at the client can be minimized by careful selection of cryptographic parameters. For example, the signature verification time is only 4% of the signature generation time when the public exponent is 17 in the RSA signature scheme.

VIII. Conclusion

MPEG-4 is the latest media stream processing standard which possesses the important "compress once, decompress many ways" property, i.e., it allows users to extract any sequence of low quality streams from a single compressed stream. In this paper, we have presented three scalable authentication schemes for MPEG-4 streams in multicast and lossy networks.

The first scheme, FAS, is the simplest, provides the maximum flexibility, and is suitable for applications requiring region cropping. However, it provides only moderate security. The second scheme, PAS, has strong security strength, but it requires the object data to be totally ordered. Therefore, PAS is the first choice for stream applications which require bit-plane based down-scaling only. The third scheme, HAS, is secure against active attacks, is flexible and has low authentication overhead.

Our schemes share the novel property of "sign once, verify many ways". That is, they allow clients to verify the authenticity of down-scaled streams extracted from a single compressed MPEG-4 object group protected with a single digital signature. We note that our group-based streaming authentication framework requires more buffer space in client devices than unprotected MPEG-4 streaming applications. How to minimize buffer space and yet still achieve efficient authenticated streaming over lossy networks is an open problem.

References

[1] *Information Technology—Coding of Audio-Visual Objects- Part 1: Systems*, ISO/IEC 14 496-1:2001.

[2] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*. Upper Saddle River, NJ: IMSC, 2002.

[3] Y. Pourmohammadi-Fallah, K. Asrar-Haghighi, and H. Alnuweiri, "Internet delivery of MPEG-4 object-based multimedia," *IEEE Multimedia*, vol. 10, no. 3, pp. 68–78, Jul.–Sep. 2003.

[4] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[5] *Information Technology—Coding of Audio-Visual Objects—Part 13: Intellectual Property Management and Protection (IPMP), Extensions*, ISO/IEC 14 496-13:2004(E).

[6] C. H. Chia and M. S. Beg, "MPEG-4 video transmission over bluetooth links," in *Proc. IEEE Int. Conf. Personal Wireless Communications*, 2002, pp. 280–284.

[7] P. Ikkurthy and M. A. Labrador, "Characterization of MPEG-4 traffic over IEEE 802.11b wireless LANs," in *Proc. 27th Annu. IEEE Conf. Local Computer Networks*, 2002, pp. 421–427.

[8] G. Kuhne and C. Kuhmnch, "Transmitting MPEG-4 video streams over the internet: Problems and solutions," *ACM Multimedia*, 1999.

[9] H. Krawczyk, "The order of encryption and authentication for protecting communications (or: How secure is SSL?)," in *Proc. CRYPTO'2001*, vol. LNCS 2139, pp. 310–331.

[10] S. Kent. (2003) IP Encapsulating Security Payload (ESP), Internet Draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-06.txt

[11] N. Achir and G. Pujolle, "Network-based adaptation for MPEG-4 multicast video delivery," in *Proc. 10th Int. Conf. Telecommunications*, 2003, pp. II1452–1457.

[12] C. Yuan, B. D. Zhu, Y. Wong, S. Li, and Y. Zhong, "Efficient and fully scalable encryption for MPEG-4 FGS," in *Proc. Int. Symp. Circuits and Systems (ISCAS)*, 2003, pp. II620–623.

[13] C. Venkatramani, P. Westerink, O. Vercheure, and P. Frossard, "Securing media for adaptive streaming," *ACM Multimedia*, pp. 307–310, 2003.

[14] Crypto++ 5.2.1 Benchmarks. [Online]. Available: http://www.eskimo.com/~weidai/benchmarks.html

[15] S. Miner and J. Staddon, "Graph-based authentication of digital streams," in *Proc. IEEE Symp. Security and Privacy*, 2001.

[16] J. M. Park, E. K. Chong, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Trans. Inform. and Syst. Security*, vol. 6, no. 2, pp. 258–285, 2003.

[17] A. Pannetrat and R. Molva. Efficient multicast packet authentication. presented at ISOC Network and Distributed System Security Symposium (NDSS). [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/03/proceedings/

[18] Y. Wu and R. H. Deng, "Content-aware authentication of motion JPEG2000 stream in lossy networks," *IEEE Trans. Consumer Electron.*, vol. 49, no. 4, pp. 792–801, 2003.

[19] R. Gennaro and R. Rohatgi, "How to sign digital streams," in *Proc. CRYPTO'97*, LNCS 1294, pp. 180–197.

[20] C. K. Wong and S. S. Lam, "Digital signatures for flows and multicasts," in *Proc. IEEE ICNP'98*.

[21] A. Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient authentication and signature of multicast streams over lossy channels," in *Proc. IEEE Symp. Security and Privacy*, 2000.

[22] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proc. CRYPTO'96*, vol. LNCS 1109, 1996, pp. 1–15.

[23] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet authentication," in *Proc. 6th ACM Conf. Computer and Communication Security*, 1999, pp. 93–100.

[24] A. Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient and secure source authentication for multicast," in *Proc. NDSS*, San Diego, CA, Feb. 2001.

[25] P. Golle and N. Modadugu, "Authenticating streamed data in the presence of random packet loss," in *Proc. NDSS*, San Diego, CA, Feb. 2001.

[26] Y. Wu, D. Ma, and C. Xu, "Efficient object-based stream authentication," in *Proc. Indocrypto 2002*, vol. LNCS 2551, pp. 354–367.

[27] C.-Y. Lin and S.-F. Chang, "Issues and solutions for authenticating MPEG video," in *SPIE Security and Watermarking of Multimedia Contents*, 1999.
[28] *Secure Hash Standard (SHS)*, FIPS Publication 180-1, 1995.
[29] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[30] "Proposed federal information processing standard for digital signature standard (DSS)," *Federal Register*, vol. 56, no. 169, pp. 42 980–42 982, 1991.
[31] C. Peng, R. H. Deng, Y. Wu, and W. Shao, "A flexible and scalable authentication scheme for JPEG2000 codestreams," *ACM Multimedia*, pp. 433–441, 2003.
[32] P. Devanbu, M. Gertz, A. Kwong, C. Martel, G. Nuckolls, and G. Stubblebine, "Flexible authentication of xml documents," in *Proc. 8th ACM Conf. Computer and Communication Security*, 2001, pp. 136–145.
[33] L. O'Connor and G. Karjoth, "Efficient downloading and updating applications on portable devices using authentication trees," in *Proc. 4th Working Conf. Smart Card Research and Advanced Application (CARDIS)*, 2000, pp. 327–344.
[34] P. Devanbu, M. Gertz, C. Martel, and S. Stubblebine, "Authentic third-party data publication," in *Proc. 14th IFIP WG11.3 Working Conf. Database Security*, vol. 201, 2001, pp. 101–112.
[35] H. H. Pang and K.-L. Tan, "Authenticating query results in edge computing," in *Proc. IEEE Int. Conf. Data Engineering*, 2004, pp. 560–571.
[36] R. C. Merkle, "A certified digital signature," in *Proc. Crypto'89*, vol. 0435, Lecture Notes on Computer Science, 1989, pp. 218–238.
[37] S. Lin and D. J. Costello Jr, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2004.
[38] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Comput. Commun. Rev.*, vol. 27, 1997.
[39] *Information Technology—Coding of Audio-Visual Objects- Part 2: Visual*, ISO/IEC 14 496-2:2003.
[40] J.-Y. Pyun, J.-H. Jung, J.-J. Shim, S.-J. Ko, and S.-H. Park, "Packet loss resilience for MPEG-4 video stream over the internet," in *Proc. Int. Conf. Consumer Electronics*, 2002, pp. 164–165.
[41] X. K. Yang, C. Zhu, Z. G. Li, X. Lin, and N. Ling, "Unequal packet loss resilience for MPEG-4 video over the internet," in *Proc. Int. Symp. Circuits and Systems*, 2003, pp. II 832–835.
[42] I. Andoh, H. Takehara, H. Nakamura, B. Lan, and H. C. Chih, "MPEG-4 video streaming over unstable broadband network environment," in *Proc. Int. Conf. Consumer Electronics*, 2002, pp. 280–281.
[43] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.* , vol. 18, no. 6, pp. 819–828, Jun. 2000.
[44] H.-F. Hsiao, Q. Liu, and J.-N. Hwang, "Layered video over IP networks by using selective drop routers," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2002, pp. I441–444.
[45] M. Bellare and D. Micciancio, "A new paradigm for collision-free hashing: Incrementality at reduced cost," in *Proc. EUROCRYPT97*, vol. LNCS 1233, 1997, pp. 163–192.
[46] M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 432–441, Mar. 2000.
[47] M. T. Goodrich, R. Tamassia, and J. Hasic, "An efficient dynamic and distributed cryptographic accumulator," in *Proc. ISC2002*, vol. LNCS 2433, pp. 372–388.
[48] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE Infocom 1999*, New York.

Yongdong Wu received the B.A. and M.S. degrees in automation control from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1991 and 1994, respectively, and the Ph.D. degree in pattern recognition and intelligent control from the Institute of Automation, Chinese Academy of Science, in 1997.

He is currently an Associate Lead Scientist with the Infocomm Security Department, Institute of Infocomm Research ($\mathrm{I}^2\mathrm{R}$), Singapore. His research interests include multimedia security, e-Business, digital right management, and network security.

Dr. Wu was awarded the Tan Kah Kee Young Inventor Award in 2004 and 2005.

Robert H. Deng received the B.Eng. degree from the National University of Defense Technology, China, in 1981, and the M.Sc. and Ph.D. degrees from Illinois Institute of Technology, Chicago, in 1983 and 1985, respectively.

He is currently Professor and Director of SIS Research Center, School of Information Systems, Singapore Management University. Prior to this, he was Principal Scientist and Manager of the Infocomm Security Department, Institute for Infocomm Research, Singapore. He has 21 patents and more than 140 technical publications in international conferences and journals in the areas of digital communications, network and distributed system security, and information security.

Dr. Deng has served as General Chair, Program Chair, and Program Committee Member of numerous international conferences. He received the University Outstanding Researcher Award from the National University of Singapore in 1999.