End-to-End Secure Delivery of Scalable Video Streams

Kianoosh Mokhtarian
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

Mohamed Hefeeda
School of Computing Science
Simon Fraser University
Surrey, BC, Canada

ABSTRACT

We investigate the problem of securing the delivery of scalable video streams so that receivers can ensure the authenticity (originality and integrity) of the video. Our focus is on recent scalable video coding techniques, e.g., H.264/SVC, that can provide three scalability types at the same time: temporal, spatial, and quality (or PSNR). This three-dimensional scalability offers a great flexibility that enables customizing video streams for a wide range of heterogeneous receivers and network conditions. This flexibility, however, is not supported by current stream authentication schemes in the literature. We propose an efficient authentication scheme that accounts for the full scalability of video streams: it enables verification of all possible substreams that can be extracted and decoded from the original stream. Our evaluation study shows that the proposed authentication scheme is robust against packet losses, adds low communication and computation overheads, and is suitable for live streaming systems as it has short delay.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—security and protection

General Terms

Security, Design

Keywords

Multimedia authentication, secure streaming, scalable video

1. INTRODUCTION

The demand for multimedia services has been steadily increasing in the past few years and is expected to grow even faster in near future [1]. Accordingly, secure delivery of multimedia content over open and generally insecure networks, e.g., the Internet, has become an important and critical concern in media distribution applications. In this paper, secure delivery refers to ensuring that the content is authentic and is not tampered with by any attacker. Various challenges need to be dealt with for this purpose. First, the authentication mechanism, which can be computationally expensive, has to keep up with the online nature of the streams. Second, media content is often distributed over unreliable channels, where packet losses are not uncommon. The authentication scheme needs to function properly even in the presence of these losses. Third, media streams can be encoded in scalable (or layered) manner to accommodate heterogeneous clients and varying network conditions. In this case, the authentication scheme has to successfully verify *any* substream extracted from the original stream. Finally, the authentication information added to the streams should be minimized in order to avoid increasing the already-high storage and network bandwidth requirements for multimedia content.

We focus on recent scalable video coding (SVC) techniques that offer great flexibility while incurring much lower overheads than traditional techniques [14]. For example, the recently standardized scalable extension of the H.264/AVC coding standard, known as H.264/SVC [11], supports adapting a video stream along three scalability dimensions: temporal, spatial, and quality (PSNR or fidelity). This three-dimensional scalability model is depicted in Fig. 1. This flexibility makes it possible to encode a video once and decode it on a wide spectrum of receiving devices, ranging from limited-capability cellular phones to high-end powerful workstations. The three-dimensional scalability model, which is more general than the previous, and much simpler, linear layered models, allows different combinations of layers along the three dimensions. Even for the same number of layers, there could be several possible *paths* through the scalability cube to achieve it [11]. Because of the many possible combinations of layers, previous authentication schemes are not directly applicable to this model. In addition, there are new coding tools employed in recent scalable coding standards, which also require new authentication techniques. For example, quality scalability in H.264/SVC is realized using the so-called Medium Grain Scalability (MGS) which was not used in previous scalable coders. To the best of our knowledge, there are no authentication schemes in the literature that can efficiently support the full flexibility of the three-dimensional scalability model.

In this paper, we design an efficient authentication scheme for video streams encoded using the general three-dimensional scalability model that allows verification of all possible substreams. The proposed authentication scheme takes into account the coding characteristics of each scalability dimension, and we show how this scheme can be applied on streams encoded with the state-of-the-art H.264/SVC coders. In addition, the proposed scheme is designed for *end-to-end* authentication of streams. In an end-to-end authentication procedure, a content provider prepares the authenticated video and sends it to receivers, possibly through a third-party Content Delivery Network (CDN) with proxy servers that

NOSSDAV'09, June 3–5, 2009, Williamsburg, Virginia, USA.

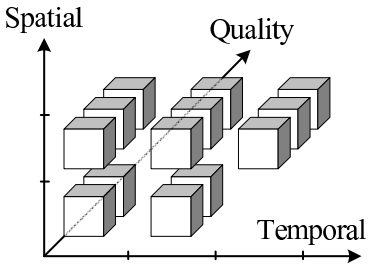


Figure 1: The three-dimensionally scalability offered by recent scalable video coders such as H.264/SVC.

may need to adapt the flexible video streams. These proxies or any other entity involved in the delivery process do not have to understand our authentication scheme, which is an important advantage of the proposed scheme. We evaluate different performance aspects of our scheme including the communication overhead, robustness against packet losses, computational cost, delay in live streaming, and buffer requirements for receivers.

This paper is organized as follows. We summarize related works in Section 2. Our authentication scheme is presented in Section 3. In Section 4, we evaluate the performance of our scheme, and we conclude the paper in Section 5.

2. RELATED WORK

The problem of authenticating video streams has been addressed by several previous works. To support adaptation of video streams, some works follow a content-based approach, whereas others consider the scalable structure of the video. In content-based methods, such as [3, 7], the general procedure is to extract a feature set from the video content and sign it. The main challenge is to have the features robust against adaptations, but fragile against malicious manipulations. In these approaches, there is no clear boundary for differentiating valid changes to the content from malicious ones, e.g., [3] relies on threshold numbers provided as input. In addition, it is not clear how significantly one can tamper with the video while preserving the feature set, e.g., [7] uses the energy distribution of I-frames as the feature set, which is not difficult to preserve while changing the content. An alternative way for making sure the video is not tampered with is that the sender embeds a *watermark* inside the video. The watermark could be a shared secret between the sender and receivers [10], or a digital signature on the video content [5]. The former case needs to trust all receivers, which is not desirable. In the latter case, the problem of deciding how to extract robust features to sign still exists.

Another approach to authenticate scalable videos is to consider their adaptation structures and authenticate their substreams. Several such techniques are proposed for classic scalable videos [4]. These videos consist of a base layer and a number of enhancement layers that progressively improve the video in terms of spatial resolution or visual quality. Authentication schemes for these one-dimensional scalable videos generally rely on two cryptographic techniques as their basis: hash chaining and Merkle hash trees. Authentication schemes based on hash chaining, e.g., [12, 17], work as follows. First, each enhancement layer of a frame is hashed and its hash is attached to its preceding layer of the same frame. The base layer hash will thus serve as a digest for all layers of the frame, i.e., the frame digest. The sequence of frames in the stream is authenticated by hash-chaining the frame digests, making a two-dimensional hash chaining scheme. Dropping higher enhancement layers has no impact on authentication of the remaining layers.

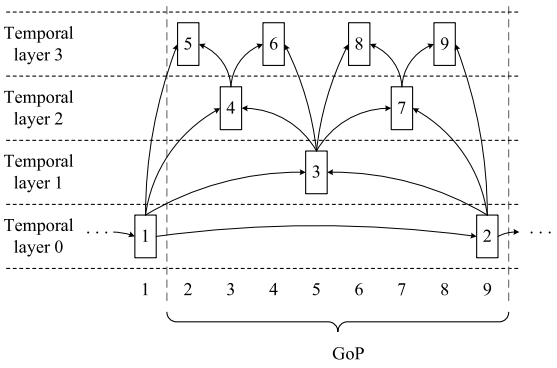


Figure 2: Hierarchical prediction structure of H.264/SVC temporal scalability. Numbers in the bottom row show the displaying order, and numbers inside frames show the coding order.

On the other hand, authentication schemes that base Merkle hash trees, e.g., [6, 15], work as follows. The enhancement layers of a video frame are hashed, and the hash values are arranged as leaves of a tree. Each interior node of this tree consists of the digest of its children. The root of the tree represents the frame digest. Due to the collision-free property of the hash function, the whole set of layers represented by the leaves is authenticated if the root of the tree is successfully verified. The sequence of frames is authenticated by building another Merkle tree over frame digests. Upon removal of some layers, a receiver may need some extra digests for verifying the remaining layers, i.e., for reconstructing the root digest of the hash tree. This means that an adapting proxy on the delivery path must understand and be compatible with the authentication scheme, which may not be desirable. Scalable authentication based on Merkle hash trees can be employed for end-to-end authentication if we embed in each layer i all information needed to authenticate the first i layers. This, however, significantly increases the communication overhead [4].

In summary, authentication techniques for scalable streams are designed for traditional simple scalable videos; they cannot support the scalability structure of modern scalable streams. For example, if applying hash-chaining for authentication of a temporal scalable stream, in addition to loss resilience issues, the temporal layers of the video stream have to be entirely kept or truncated, which limits the flexibility of the stream; it can no longer be adapted to any arbitrary frame rate. As another example, applying previous techniques to authenticate quality enhancement packets may result in unverifiability of some of the received packets, as we see in Section 3.3.

3. AUTHENTICATION OF H.264/SVC VIDEO STREAMS

3.1 Brief Overview of H.264/SVC

The recently standardized H.264/SVC [11] video coding technique adds scalability to the state-of-the-art H.264/AVC video coding technique. In addition to generating highly flexible streams, H.264/SVC significantly outperforms previous scalable video coding techniques in terms of coding efficiency [14]. H.264/SVC supports temporal, spatial, and quality scalability at the same time. Temporal scalability is achieved by employing a hierarchical prediction structure among video frames belonging to the same Group-of-Pictures (GoP), as shown in Figure 2. In the spatial scalability of SVC, a spatial layer s of a frame can be predicted from the s-th spatial layer of some other frames, as well as lower spatial layers

in its own frame. For providing quality scalability, there are two two different possibilities. The first one is to follow the spatial scalability structure, but assign the same resolution and different quantization parameters to layers. In this way, a *Coarse-Grained* Scalable (CGS) video is obtained. A finer granularity can be provided by the second possibility, which uses Medium-Grained Scalability (MGS) coding to partition a CGS layer into multiple MGS layers. A stream can not only be truncated at each CGS or MGS layer, but some packets of an MGS layer can be discarded as well. H.264/SVC allows up to 7 temporal, 8 spatial and 16 quality layers.

In H.264/SVC, the coded video data and other related information are organized into Network Abstraction Layer (NAL) units. Each NAL unit, which we alternatively refer to as a video packet, has a *temporal_id*, *spatial_id*, and *quality_id* value, which identify the layers the NAL unit belongs to. NAL units can be Video Coding Layer (VCL) NAL units, which contain the coded video data, or non-VCL NAL units, containing associated additional information. Supplemental Enhancement Information (SEI) NAL units are a type of non-VCL NAL units that are not required for decoding the video. They are used to carry auxiliary information to assist the decoding process. We exploit SEI NAL units (NAL unit type 6) for carrying the authentication information in an SVC-compatible manner. For attaching some information to a specific layer, we embed these information to a SEI NAL unit having the same temporal/spatial/quality identifiers in its header as the target layer.

3.2 Overview of the Proposed Scheme

At a high level, the authentication mechanism operates as follows. First, the content provider prepares the additional information needed for verification, and attaches it to the stream. Each receiver either receives the whole stream or a subset of the layers, along with the corresponding authentication information. The task of substream extraction may be carried out by third-party proxies belonging to the delivery network. The authentication information is transparent to these proxies; it is attached to specific NAL units in a format-compliant manner, as pointed earlier.

Generating the authentication information. An SVC stream is a sequence of GoPs. Each GoP consists of a number of video frames, each of which belongs to a certain temporal level. Each frame, in turn, contains multiple spatial layers. A spatial layer then includes a few CGS quality layers, each one possibly partitioned into several MGS layers. Each MGS layer too can be divided into multiple video packets. According to this structure, the server prepares the authenticated video using the following steps:

1. *Authenticate CGS and MGS quality layers:* Within each spatial layer of each frame, quality layers are first authenticated. This process needs to take into account that quality layers in SVC, unlike previous scalable videos, can be extracted in many possible ways. Moreover, MGS quality layers are no more atomic units of data: a single MGS layer can be truncated at packet level [11]. The computed authentication information is attached back to the quality layers, and the hash value of these information forms the *spatial layer digest*.
2. *Authenticate a video frame:* In this step, spatial layers and their digests are authenticated, authentication information is attached to the layers, and one hash value is created as the *frame digest*. This step should consider that the dependency among layers is not simply linear, as it was in previous scalable videos. Moreover, authentication information of quality and spatial layers needs to be protected against loss with a controlled amount of communication overhead.
3. *Authenticate a GoP:* For authenticating each GoP, its frame digests are correlated in a certain way and distributed among the frames themselves. This is done according to the temporal levels in the GoP so that adaptation of the stream to any arbitrary frame rate does not affect our scheme. The result of this step is a *GoP digest* for each GoP.
4. *Authenticate the whole stream:* The stream, which is a sequence of GoPs, is authenticated by dividing the sequence into *blocks*, each containing a small number of GoPs. Out of GoP digests of each block, a *block digest* is obtained, which is digitally signed. The authentication information of the block is attached to its GoPs in a loss-resilient manner.

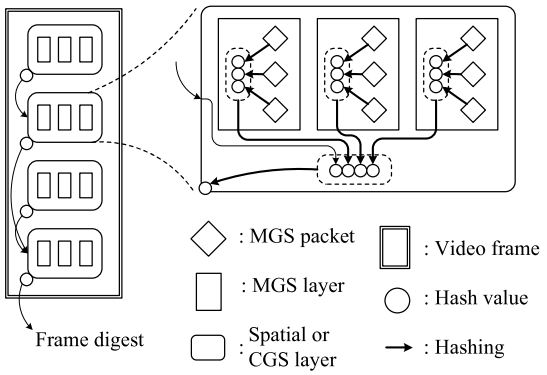


Figure 3: Authenticating a video frame.

Each of the above digests hides all the scalability details of the digested unit of data. For example, a frame digest hides spatial and quality scalability, and makes the frame a transparent unit for the next step of authentication. Given any valid subset of packets of a frame and their authentication information, their authenticity can be verified if and only if the frame digest is successfully received and verified; similarly for spatial layer, GoP, and block digests.

Verifying substreams. The verification process proceeds in the same way as generating the authentication information. Given a valid substream and its authentication information, a receiver recomputes spatial layer, frame, GoP, and block digests from the received video. In case of any mismatch between these digests and the digests provided by the server in the substream, the mismatching part of data, such as a video frame, is marked as unauthentic and is discarded. The remaining part of the received substream is authentic if and only if the digital signature of the corresponding block is successfully verified. In this way, all video packets of any valid substream can be authenticated.

Handling packet losses. Packet losses in video transmission can be tolerated to some extent using techniques such as error concealment and interleaved packetization [13]. However, loss of the authentication information carried by a layer has a serious effect: some higher layers cannot be verified and thus cannot be used, even though they are received and could have been decoded and used. We therefore need to appropriately protect the authentication information against loss. If the video is being transmitted over the Internet, where bursts of packets can be lost, it is a common practice to distribute the video data over network packets in an interleaved manner [13], which changes the loss pattern from bursty to random. Relying on such packetization technique, we assume packet losses have a random pattern.

3.3 Details of the Proposed Scheme

Authentication of Quality Layers. Quality scalability can be provided in SVC by encoding one or more CGS layers, and partitioning each CGS layer into multiple MGS layers. As discussed

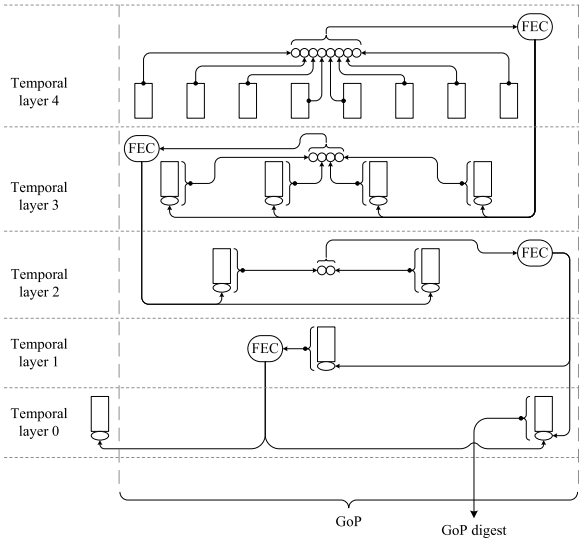


Figure 4: Authenticating frames of a GoP. Small solid circles at the beginning of some links represent hash operations.

earlier, CGS layers are encoded the same way as spatial layers, and follow the same dependency structure. We thus treat CGS layers as spatial layers; when dealing with authentication of quality scalability, we only consider MGS layers within a spatial or CGS layer.

In traditional quality-scalable videos, the hash of each enhancement layer is attached to the previous layer, providing the hash of the base layer as the video frame digest. However, in MGS quality scalability, it is possible that some video packets of a higher MGS layer be kept while some packets from a lower MGS layer are discarded, depending on the extraction process [2]. Consequently, simply hash-chaining the MGS layers will not work: the hash chain can be broken and some MGS layers can be left non-authenticatable. Since the H.264/SVC standard does not dictate any specific extraction process, we cannot rely on any specific ordering of MGS packets. Thus, we form the authentication information of MGS packets in a two-level hierarchy as shown in Figure 3. For video packets of each spatial (or CGS) layer s, the procedure is as follows. First, packets of each MGS layer q are hashed using a secure hash function $h(\cdot)$, and their hashes are concatenated. Denote this concatenation by $F_{s,q}$ and the number of MGS layers in the s-th spatial layer by Q_s. Then, $F_{s,q}$ values are hashed, $H_{s,q} = h(F_{s,q})$, and concatenated as $F_s = H_{s,0}||\ldots||H_{s,Q_s}$ so that $H_s = h(F_s)$ is obtained as the digest of the spatial layer. Moreover, each $H_{s,q}$ is attached to the q-th MGS layer.

Authentication of a Video Frame. In spatial (and CGS quality) scalability, the dependency structure among spatial layers of a frame is not necessarily linear. Rather, it is a Directed Acyclic Graph (DAG). For example, layer 5 can use layers 1 and 3, rather than 4, as reference layers. Therefore, attaching the digest of each spatial layer to the previous one, as in hash chaining, would not work. We thus attach each spatial layer digest H_s to its highest reference layer. This is shown in Figure 3. The digest of the lowest spatial layer represents the digest of the video frame.

To protect the authentication information of quality and spatial layers against loss, we need to add redundancy. The common technique for this purpose is the use of Forward Error Correction (FEC) codes. However, since there can be several quality and spatial layers, computational cost of performing many FEC operations per each video frame can be too high. Therefore, we replicate the authentication information of each quality/spatial layer in two or more packets, rather than FEC-coding these information. The number of copies of the authentication packet can balance a tradeoff between loss tolerance and communication overhead. We will show in the Evaluation section that only 2 copies are enough for resisting against reasonable loss ratios.

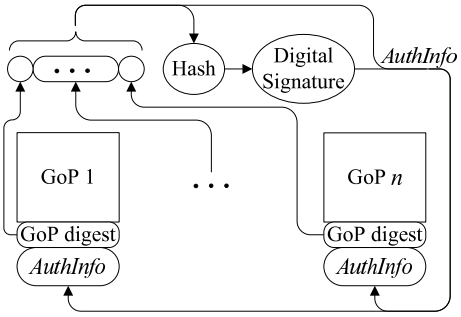
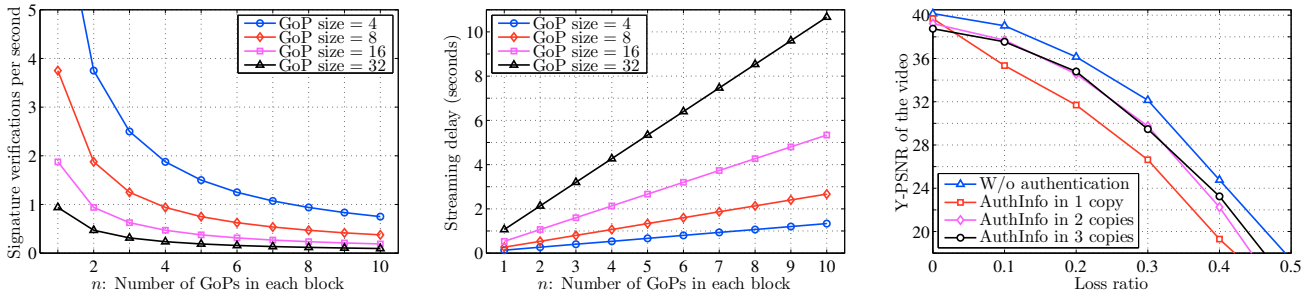


Figure 5: Authentication of a sequence of GoPs.

Authentication of a GoP. We take care of temporal scalability using frame digests provided in the previous step. In SVC temporal scalability, a common practice for frame rate adaptation is to drop all frames of temporal layers beyond a certain level, say T, and possibly a fraction of frames at level T. Therefore, we embed the digests of each temporal layer in the frames of the preceding temporal layer, as depicted in Figure 4; GoPs with non-dyadic structure can also be authenticated in a similar manner. In this scheme, frame digests of temporal level T are first concatenated. Denoting the number of frames in level T by n_T, this concatenation is divided into k_T ($k_T \leq n_{T-1}$) pieces, FEC-coded into n_{T-1} pieces, and distributed over the n_{T-1} frames of level $T-1$. In this way, adaptation of the video to any frame rate will not affect our authentication mechanism. At the receiver side, authentication information of any k_T out of n_{T-1} frames of level $T-1$ leads to successful verification of all frames at level T. Hence, the value of k_T, which is an input to our scheme, depends on the expected loss resilience. With a value of $k_T = \alpha_{T-1} n_{T-1}$, loss of the authentication information of up to $(1-\alpha_{T-1})\, n_{T-1}$ frames of the n_{T-1} frames at temporal layer $T-1$ can be tolerated. Figure 4 also shows how the GoP digest is obtained, whose authenticity is necessary and sufficient for authentication of any valid subset from the GoP.

Authentication of a Sequence of GoPs. So far we have seen how to authenticate and obtain a single GoP digest out of each GoP. No GoP can be dropped from the stream for adaptation. Thus, a sequence of GoPs can in general be thought of as a stream of data packets. Accordingly, we first consider to authenticate the sequence of GoPs by applying data stream authentication techniques.

To authenticate a stream of data packets, the common practice, e.g., [8, 9], is to divide packets of a stream into blocks of size n packets, and designating one digital signature for each block [4]. This will amortize the signature size and the computational cost of signature verification over several packets. We follow the same procedure for preparing the authentication information of a block of GoPs. Various methods are proposed for distributing the authentication information of a block in its packets in a way to best resists against bursts of packet losses. The cost of this achievement is some delay, which is especially important for live streaming, and some buffering requirement for receivers [4], since a receiver has to receive almost a complete block before being able to verify any of its packets. However, a sequence of GoPs has different characteristics than a sequence of data packets: it has a low rate, each GoP



(a) signature verifications per second. (b) Streaming delay. (c) Loss resilience of our scheme.

Figure 6: Computation cost, delay, and loss resilience of the proposed authentication scheme.

is very large compared to a packet, and GoPs are not likely to be lost in bursts, since bursts of loss have short periods [16]. Taking advantage of these considerations, we authenticate each block of GoPs as shown in Figure 5. Compared to applying classic packet stream authentication techniques for authenticating the sequence of GoPs, this method has the following benefits: (i) it is more robust against loss, since receiving the authentication information of any of the GoPs is sufficient for verifying the whole block, (ii) it incurs lower delay, because a receiver does not have to wait for receiving almost a complete block (as in [8, 9]), and (iii) it requires only negligible receiver buffer; one or very few GoPs. Our method is also efficient in terms of loss tolerance and communication overhead, as we analyze shortly.

4. PERFORMANCE EVALUATION

We use simulations to evaluate the performance of our scheme in terms of computational cost, delay, buffer requirements for receivers, loss tolerance, and communication overhead. As described in Section 2, we are not aware of other authentication schemes in the literature designed for scalable video streams that support the flexible, three-dimensional, scalability. Previous authentication schemes are not applicable to such streams, since they cannot authenticate all their possible substreams. Hence, comparing our scheme against them does not make sense.

Simulation Parameters. We simulate transmission of a video stream over a channel with frequent packet losses. The transmission packet size is assumed 1 KB. The considered video stream has a bitrate of 1 Mbps, and provides a relatively high quality of 40 dB in terms of Y-PSNR. It has 30 frames per second, is at CIF (352×288) resolution, and consists of 4 temporal layers (a GoP size of 8), 1 spatial layer, and 2 CGS layers. The CGS enhancement layer is partitioned into 3 MGS layers, each consisting of multiple quality refinement packets. A high flexibility is provided by the considered video since a subset of packets of each layer can be discarded. We employ SHA-1 as the hash function (20-byte hashes), and RSA as the digital signature scheme (128-byte signatures) due to its inexpensive verification.

Computation Cost. This is the most important performance factor of an authentication scheme. If some receivers cannot afford the computations needed by the scheme, they cannot verify the video at all —we assume the server is powerful enough for providing the authenticated stream in real-time. We assume only a small fraction of receivers' CPU (5–10%) is available for authentication operations, the dominant of which is digital signature verification. We neglect hashing and FEC coding costs as they can be performed very fast [4]. Figure 6(a) depicts the number of signature verifications needed per second for different values of n, the number of GoPs in a signed block. The value of n can balance a tradeoff between delay and computation cost. Assuming that one to two signature verifications per second are easily affordable by nowadays limited-capability video playback devices [4], Figure 6(a) shows that gathering only $n = 5$ GoPs in each block suffices for having the authentication operations affordable by all receivers.

Delay and Buffering Requirements. When streaming live content, the delay is in proportion to the block size. Figure 6(b) depicts the delay caused by the authentication mechanism for different values of n. In our case with $n = 5$ and a GoP size of 8, the delay is less than 2 seconds, which is quite acceptable. Moreover, a value of $n = 5$ indicates that in the worst case, where the authentication information of the first four GoPs of a block are lost, the receivers need to buffer 5 GoPs, which needs a small buffer only (< 250 KB). Note that this represents the buffering required by the authentication scheme; the streaming application may already be buffering a few seconds of video data before playing back, which can be utilized by the authentication mechanism as well.

Robustness Against Loss. Packet losses can negatively impact an authenticated video in two ways. First, some video packets can be lost. Second, some video packets, although received, can be unusable as they cannot be authenticated: (i) their authentication information is lost, or (ii) these packets may be authenticated via other packets, where they or their authentication information may be lost. Therefore, authentication may amplify the effect of losses. We show that our scheme does not suffer from these issues. We suppose that a receiver tries to conceal the error caused by loss of a packet. We assume a simple error concealment model that recovers loss/unverifiability of packets of a frame up to a certain ratio, and drops the frame if the threshold is exceeded. The Y-PSNR quality of a frame is reduced according to the lost packets. A lost frame is replaced by its closest frame at equal or lower temporal levels. If the replacement candidate is also lost, the replacement frame's replacement is considered. If the victim frame is used for decoding of some other frames, their Y-PSNR is also reduced similarly to the victim frame. The "W/o authentication" curve in Figure 6(c) depicts the result of the concealment model we assumed.

As discussed earlier, the authentication information of quality and spatial layers is replicated in a few, say k, copies for protection against packet loss. Figure 6(c) shows the effect of loss on the authenticated video for different values of k when receiving the full stream. As the figure illustrates, the quality of the video is slightly

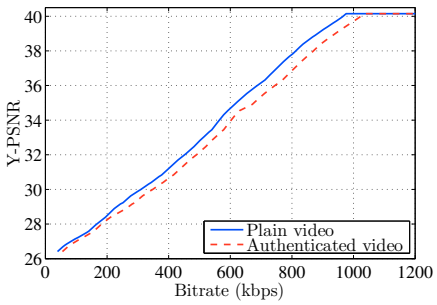


Figure 7: Communication overhead of our scheme.

reduced by increasing k (see the qualities for a loss ratio of 0) due to the communication overhead of replication, which we analyze shortly. The first finding by Figure 6(c) is that our authentication scheme increases the impact of loss only marginally: as the loss ratio increases from 0 to higher values, the gap between the plain (unauthenticated) video and the authenticated video with $k = 2$ or 3 increases negligibly. Thus, the gap between them is only the result of the overhead, which is still reasonable (about 1 dB in Y-PSNR). As a second finding, Figure 6(c) also helps us determine how many copies the authentication information packet should be sent in. According to Figure 6(c), $k = 1$ results in high sensitivity to loss, $k = 2$ and 3 are suitable for reasonable loss ratios (10–30%), and $k = 3$ becomes dominant as the loss ratio grows higher, since the tolerance it provides will then worth its overhead.

Communication Overhead. Figure 7 depicts the communication overhead of our scheme ($n = 5$, $k = 2$). In this figure, the rate-quality curves of the video before and after adding the authentication information are plotted. This overhead consumes a portion of bandwidth and causes a lower-rate video, and thus a lower quality, be received. However, this quality gap is not significant; up to 1 dB in Y-PSNR in Figure 7. The majority part of the authentication information are hash values of the quality packets, since there are several of them in each video frame. Figure 7 shows that our two-level hierarchy for authenticating MGS layers limits the amount of communication overhead for low-bitrate substreams.

5. CONCLUSION

In this paper, we have studied the end-to-end authentication problem of modern scalable video streams, which offer the flexibility of extracting different substreams without significantly decreasing the coding efficiency. This kind of scalable streams is desirable to support the increasingly heterogeneous population of clients receiving multimedia content. We developed an authentication scheme for these streams that supports their full scalability. Our evaluations show that our scheme is robust against reasonable packet loss rates (10–40%), has low communication overhead, incurs negligible computational cost, adds only a short (1–2 second) delay, and requires no significant buffering ($<$ 1 MB) by receives, unlike many of the previous schemes for traditional scalable videos [4].

It should be noted that if a stream is encoded to be highly flexible and fine-granular, the communication overhead of our authentication scheme can be non-negligible. Our ongoing work is to control this and guarantee a low amount of overhead for any stream.

6. REFERENCES

[1] Global IPTV market analysis (2006-2010). Technical report, RNCOS, August 2006.

[2] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux. Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17:1186–1193, September 2007.

[3] P. Atrey, W. Yan, and M. Kankanhalli. A scalable signature scheme for video authentication. *Multimedia Tools and Applications*, 34:107–135, July 2007.

[4] M. Hefeeda and K. Mokhtarian. Authentication schemes for multimedia streams: Quantitative analysis and comparison. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2009. Accepted to appear.

[5] R. Iqbal, S. Shirmohammadi, A. El-Saddik, and J. Zhao. Compressed-domain video processing for adaptation, encryption, and authentication. *IEEE Multimedia*, 15(2):38–50, April 2008.

[6] R. Kaced and J. Moissinac. Multimedia content authentication for proxy-side adaptation. In *Proc. of International Conference on Digital Telecommunications (ICDT'06)*, Cap Esterel, Cote d'Azur, France, August 2006.

[7] C. Liang, A. Li, and X. Niu. Video authentication and tamper detection based on cloud model. In *Proc. of Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'07)*, volume 1, pages 225–228, Splendor Kaohsiung, Taiwan, November 2007.

[8] A. Pannetrat and R. Molva. Efficient multicast packet authentication. In *Proc. of Network and Distributed System Security Symposium (NDSS'03)*, San Diego, CA, Feb. 2003.

[9] J. Park, E. Chong, and H. Siegel. Efficient multicast stream authentication using erasure codes. *ACM Transactions on Information and System Security*, 6(2):258–285, May 2003.

[10] S. Park and S. Shin. Combined scheme of encryption and watermarking in H.264/Scalable Video Coding (SVC). In *New Directions in Intelligent Interactive Multimedia*, volume 142 of *Studies in Computational Intelligence*, pages 351–361. Springer, September 2008.

[11] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[12] D. Skraparlis. Design of an efficient authentication method for modern image and video. *IEEE Transactions on Consumer Electronics*, 49(2):417–426, May 2003.

[13] S. Wenger. H.264/AVC over IP. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):645–656, July 2003.

[14] M. Wien, H. Schwarz, and T. Oelbaum. Performance analysis of SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1194–1203, September 2007.

[15] Y. Wu and R. Deng. Scalable authentication of MPEG-4 streams. *IEEE Transactions on Multimedia*, 8:152–161, February 2006.

[16] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modeling of the temporal dependence in packet loss. In *Proc. of IEEE INFOCOM'99*, volume 1, pages 345–352, New York, NY, March 1999.

[17] H. Yu. Scalable streaming media authentication. In *Proc. of IEEE International Conference on Communications(ICC'04)*, volume 4, pages 1912–1916, Paris, France, June 2004.