5-2016

Towards Secure Online Distribution of Multimedia Codestreams

Swee Won LO
Singapore Management University, sweewon.lo.2009@phdis.smu.edu.sg

Towards Secure Online Distribution of Multimedia Codestreams

by

LO Swee Won

Submitted to School of Information Systems in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Robert Huijie DENG (Supervisor / Chair)
Professor of Information Systems
Singapore Management University

Xuhua DING (Co-supervisor)
Associate Professor of Information Systems
Singapore Management University

Yingjiu LI
Associate Professor of Information Systems
Singapore Management University

Yongdong WU
Senior Scientist and System Security Lab Head
Infocomm Security Department
Institute for Infocomm Research, Singapore

Singapore Management University
2016

Towards Secure Online Distribution of Multimedia Codestreams

LO Swee Won

Abstract

Multimedia codestreams distributed through open and insecure networks are subjected to attacks such as malicious content tampering and unauthorized accesses. This dissertation first addresses the issue of authentication as a mean to integrity-protect multimedia codestreams against malicious tampering. Two cryptographic-based authentication schemes are proposed to authenticate generic scalable video codestreams with a multi-layered structure. The first scheme combines the salient features of hash-chaining and double error correction coding to achieve loss resiliency with low communication overhead and proxy-transparency. The second scheme further improves computation cost by replacing digital signature with a hash-based message authentication code to achieve packet-level authentication and loss-resiliency. Both schemes are robust to transcoding, i.e., they require only one-time authentication but allow verification on different transcoded versions. A comprehensive analysis is performed on the proposed schemes in comparison to existing work in terms of their authentication and verification delays, communication overhead, and buffer sizes needed for authentication/verification.

Scalable video codestreams encoded by the H.264/SVC standard are made up of frames with spatial and quality layers while each frame belongs to a specific temporal layer. Taking into account the dependency structure of an H.264/SVC codestream, a secure and efficient cryptographic-based authentication scheme that is fully compatible with such a structure is proposed. By integrating the temporal scalability structure with a combination of double error correction coding and packet replication techniques, the proposed scheme is highly loss-resilient with a low communication overhead under burst loss condition. Performances of the pro-

posed scheme under different encoding settings are further analyzed and the results showed that the proposed scheme outperforms an existing scheme in terms of its loss-resiliency. The proposed scheme also exhibits low authentication and verification delays, which is an important performance factor for real-time multimedia applications.

The third work in this dissertation studies the security of content-based authentication for non-scalable video codestreams. Based upon the video coding concept, it is shown that existing transform-domain content-based authentication schemes exhibit a common design flaw, where the transform-domain feature extracted is not sufficient to represent the true semantic meaning of the codestreams. Consequently, although the schemes are able to detect semantic-changing attacks performed in the pixel domain, they are unable to detect attacks performed in the transform domain. A comprehensive discussion on how the flaw can be exploited by manipulating transform domain parameters is presented and several attack examples are demonstrated. In addition, the concept behind attacks that manipulate the transform-domain header parameters and the conditions of the attacks, given the attacker's desired attack content, are discussed in depth.

Finally, the issue of access control as a mean to regulate unauthorized accesses to protected codestreams is studied. For generic scalable codestreams, a secure and efficient access control scheme is presented, where symmetric encryption is used to protect the codestreams, and attribute-based encryption is used to disseminate access keys to users. We further extend the scheme to address access control for H.264/SVC codestreams. The proposed schemes are secure against collusion attack and employ access keys generation hierarchy that is fully compatible to the dependency structures of generic and H.264/SVC codestreams, respectively. As a result, they are efficient in the way that each user needs to maintain only a single access key regardless of the number of layers he/she is entitled to access. The proposed schemes also eliminate the use of an online key distribution center by employing attribute-based encryption for access keys dissemination.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my supervisors Professor Robert Deng and Associate Professor Xuhua Ding for their guidance and care. I'm immensely lucky to be able to meet them, and then work with their guidance throughout my course of study. I would also like to thank my dissertation committee members Associate Professor Yingjiu Li and Dr. Yongdong Wu for their kind and constructive feedback on this work. I would also like to express gratitude to my family for being so supportive and encouraging, and a great friend Lucia for being there whenever things get tough. Finally, I would like to thank our PhD program administrators, Chew Hong and Pei Huan, for the help in handling many of our issues and the occasional uttered encouragement whenever we met in or out of school.

Dedication

I dedicate my dissertation work to my beloved mom, brother and Li Feng.

List of Publications

Conference Papers

Y. Zhao, S.-W. Lo, R. H. Deng and X. Ding. An improved authentication scheme for H.264/SVC and its performance evaluation over non-stationary wireless mobile networks. In *Proceedings of the 6th International Conference on Network and System Security*, China, 2012.

Z. Wei, R. H. Deng, J. Shen, Y. Wu, X. Ding and S.-W. Lo. Technique for authenticating H.264/SVC streams in surveillance applications. In *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, San Jose CA, 2013.

S.-W. Lo, Z. Wei, X. Ding and R. H. Deng. Generic attacks on content-based video stream authentication. In *Proceedings of the 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, China, 2014.

S.-W. Lo, Z. Wei, X. Ding and R. H. Deng. Generic attacks on content-based video stream authentication (demo paper). In *Proceedings of the 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, China, 2014.

S.-W. Lo, Z. Wei, R. H. Deng and X. Ding. On Security of Content-based Video Stream Authentication. In *Proceedings of the 20th European Symposium on Research in Computer Security (ESORICS)*, Austria, 2015.

Z. Wei, S.-W. Lo, Y. Liang, T. Li, J. Shen and R. H. Deng. Automatic accident detection and alarm system. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Austria, 2015.

Journal Papers

R. H. Deng, X. Ding and S.-W. Lo. Efficient Authentication and Access Control of Scalable Multimedia Streams over Packet-Lossy Networks. *Security and Communication Networks*, 7(3), 2013, pp. 611-625.

Y. Zhao, S.-W. Lo, R. H. Deng and X. Ding. Technique for Authenticating H.264/SVC and its Performance Evaluation over Wireless Mobile Networks. *Journal of Computer and System Sciences (Special Issue)*, 80(3), 2014, pp. 520-532.

Chapter 1

Introduction

Multimedia formats such as images and videos are commonly used as the main visual media in many online applications such as news portals as well as digital advertising, where in the latter they are regarded as the most important elements that can effectively convey an intended brand story [12]. Apart from that, images and videos obtained from surveillance feeds or from a vehicle's dashboard camera can also be used as evidences in court in case of disputes. In the telemedicine industry, patient's medical images (such as X-rays) are stored in an online medical system so that personnels across different departments/hospitals can perform timely diagnosis at anytime and anywhere, whereas videos are also being rapidly adopted to provide live consultations for patients living in rural areas [1]. Images and videos are also the most prominently shared multimedia formats in many social media platforms as they are able to generate a high level of interactions among users [20]. Without a doubt, online multimedia applications are slowly revolutionizing the way businesses and day-to-day livings are conducted.

In retrospect, the proliferation of online multimedia applications also raises several security issues. One of the main security concerns is the authenticity of multimedia codestreams. Authentication addresses two main questions: who the sender is, and whether the multimedia codestream has been maliciously tampered while in transit. Being able to answer these questions is important because malicious tam-

pering on multimedia codestreams could be motivated by numerous commercial or political purposes, or with the intention to evade law (e.g., attacks on surveillance codestreams). Recently, a case where a police vehicle dashboard video showing arrest of a woman allegedly assaulted the police officer is suspected to be edited before it was released [38]. An incident where a video of the Greek Finance Minister making a rude gesture while addressing relationship between Greece and Germany has created diplomatic controversy; till now, there are no solid evidences on whether the video has been doctored [6]. Such incidents would not only result in social and political unrests, it could also hamper the future adoption of many useful online multimedia applications. It is thus imperative that an appropriate authentication mechanism is in place to thwart malicious tampering on online multimedia codestreams.

Another threat is unauthorized content access. Access control involves authorizing legitimate users with appropriate privileges to access a certain resource while denying access from illegal users. In many commercial multimedia applications such as video-on-demand, providers may publish free preview of a movie in low resolution and offer packages of different privilege levels for users who are interested to purchase the movie in higher resolutions/frame rates. Different users may then subscribe to different privilege levels, each granting access right to different codestream qualities. In such setting, a dishonest user may attempt to access unauthorized codestream content of quality beyond his or her access rights. For the provider, it is hence vital to ensure that only legitimate users have access to the multimedia codestreams, and that each authorized user can only access codestream content of quality corresponding to the privilege level he or she subscribes to. Furthermore, two or more users subscribing to different privilege levels must not be able to collude to obtain access rights to a higher privilege level. To achieve this, a collusion-free access control scheme must be in place to deny accesses from illegal users while authorizing legitimate users with appropriate privilege levels to access the corresponding resources.

The main focuses of this dissertation are on the issues of authentication and access control for video codestreams. Compared to images, videos are finding their ways into much wider practical applications. For example, in the digital advertising industry, online video has the highest click-through rate compared to all other digital advertising format [33] and video is about to emerge as the main marketing method for businesses around the world [26]. Home networked video surveillance is also expected to increase from 29% in 2014 to 49% by 2016 [83] whereas 71% of overall retailers are set to adopt networked video surveillance as a part of their loss prevention strategy [67]. In the telehealth industry, live video used for consultation or to convey medical records has become the most frequently covered insurance service in the United States [1]. It is predicted that by the year 2019, million minutes of video content is to cross the network in every second [11]. On the other hand, the concepts and performance requirements for the authentication and access control schemes proposed in this dissertation is also applicable to the protection of image codestreams such as those encoded by the JPEG and JPEG2000 standards. More specifically, cryptographic-based approaches in Chapters 3, 4 and 6 works by identifying the dependency structure in a scalable codestream and designing efficient and proxy-transparent security schemes that preserve the codestream dependency structure. The content-based authentication approach in Chapter 5 presents an important concept, which is the identification of a "feature" that can fully characterize the interdependent relationship between the transform-domain coefficients and header parameters of a multimedia codestream in order to thwart any attempt to manipulate the multimedia content. Since a video is essentially a sequence of still images with an additional temporal domain, the methods and concepts proposed in this dissertation can serve as references for the authentication/access control for other multimedia formats such as images.

1.1 Problem Formulation

It is nonetheless a challenging problem to realize authentication and access control in an online multimedia distribution framework. Online multimedia applications prioritize quality of service but security introduces additional computation and communication overhead on top of those required by the multimedia applications. Hence, a security scheme designed for multimedia codestreams has to be as *lightweight* - in terms of computation costs and communication overhead - as possible. In addition, it should take up a small amount of *buffer size* to support real-time applications, especially on mobile devices.

Multimedia codestreams distributed through the network are also subjected to packet loss due to transmission errors and network congestion. For this reason, various error concealment techniques are incorporated in multimedia coding standards in order to optimize viewing experience in the event of loss. In retrospect, cryptographic algorithms are extremely sensitive to packet loss. To avoid compromising viewing experience due to a high percentage of received but unverifiable packets, it is imperative that the security scheme is also *loss-resilient*. A common solution is to add redundancies on the security data using error correction code, but the amount of redundancies incurs additional communication overhead. Care must then be taken to integrate error correction coding techniques with cryptographic algorithms to minimize the communication overhead needed to cater for packet loss.

In a multimedia distribution framework, a multimedia codestream can be encoded and distributed as either a **non-scalable** or a **scalable** codestream. A non-scalable codestream such as one produced by the H.264/AVC (Advanced Video Coding) [91] standard is optimally encoded at a pre-specified bit-rate and consists of intra- and inter-predicted frames, where intra-predicted frame is encoded using information in the frame itself and thus is more essential for optimal multimedia representation compared to inter-predicted frames. On the other hand, a scalable codestream such as one encoded using the H.264/SVC (Scalable Video Coding)

[75] standard is a multi-layered codestream consisting of a (mandatory) base layer that decodes to a coarse multimedia representation, and one or more enhancement layers, each improving the decoded base layer in terms of either frame rate, resolution or SNR (Signal-to-Noise Ratio). These internal dependency structures of a codestream should be taken into consideration while designing the security scheme so as not to compromise users' viewing experience.

Another concern in online distribution of multimedia codestreams is the problem of transferring multimedia data as a continuous codestream so that a device can start playback before receiving the codestream in full. This effort is made more challenging when receiving devices are heterogeneous in terms of computation capabilities and bandwidths of the network they are connected to. To address this, bit-rate transcoding operation adapts a codestream to a bit-rate optimum for a receiving device and the type of network it is connected to. Different transcoding techniques are applied depending on the type of codestream. As a non-scalable codestream is encoded at a pre-specified bit-rate, transcoding technique involves re-encoding the codestream using a new quantization step size. In retrospect, a scalable codestream is transcoded by discarding a subset of enhancement layers until the codestream bit-rate is optimum for the receiving device. However, since a transcoded codestream is different from the original codestream, security data computed on the original codestream would be invalidated. Hence, the security scheme should be designed such that it is *robust to transcoding* while is able to distinguish between transcoding (i.e., legitimate changes) and malicious tampering.

To liberate the multimedia source from the heavy workload of catering to a large population of heterogeneous users, transcoding operation is commonly performed by one or more intermediate proxies. Each proxy may simultaneously serve a large user population within a specific geographical area. It is thus highly desirable that the security scheme is *proxy-transparent* such that a proxy needs not be aware of the underlying security mechanism during transcoding; this can effectively relieve the proxies from additional operational overhead.

1.2 Online Multimedia Distribution Framework

The online multimedia distribution framework considered in this dissertation consists of a multimedia source, a set of proxies and a group of heterogeneous end users. The source produces a protected (scalable/non-scalable) multimedia codestream and forwards it to the proxies. The proxies are responsible for transcoding according to users' preferences and/or network conditions, and for delivering the transcoded codestream to the end users. When a user receives a transcoded codestream, the user performs decryption and/or authenticity verification on the received content. The proxy-user network may subject to packet loss due to transmission errors or traffic congestions.

1.3 Dissertation Overview

This dissertation studies the issues of authentication as well as access control in online distribution of video codestreams. The codestreams are distributed over a source-proxy-user network as described in Section 1.2 and they can be encoded in either scalable or non-scalable forms. Taking into account the respective transcoding techniques, this dissertation studies and proposes secure and efficient authentication/access control schemes that are lightweight with low buffer requirement, high resiliency to loss, robust to transcoding and proxy-transparent.

1.3.1 Authentication for Multimedia Codestreams

The two general approaches for multimedia codestream authentication are cryptographic-based authentication and content-based authentication [89]. Cryptographic-based authentication schemes make use of cryptographic techniques such as hash function and digital signature algorithm to compute authentication data on a multimedia codestream. To verify its authenticity, a user recomputes the hash of the multimedia codestream and verifies it against the digital signature.

The main advantages of cryptographic-based authentication are that the security of cryptographic algorithms are well-established and it can provide strict verification without ambiguity.

Cryptographic-based solutions for authenticating non-scalable multimedia codestreams without considering the event of transcoding is a well-studied problem (e.g., the works of [28] and [31]). However, a transcoded non-scalable codestream cannot be correctly verified using authentication data which is computed on the original codestream. A naive solution is to pre-encode a codestream into multiple copies of different bit-rates and compute the authentication data for each copy. However, this is not only storage and computationally intensive, it is also not scalable. Another solution is for the source to delegate power to proxies (using cryptographic techniques such as sanitizable signature [5]) so that they can compute valid authentication data for transcoded codestreams (e.g., the work of [16]). However, this solution is not proxy-transparent and the verification of sanitizable signature is computationally intensive for low power devices. There are also security risks as semi-trusted proxies are given the authorization to "generate" an authenticated codestream on the source's behalf.

On the other hand, a content-based authentication scheme works by extracting a *feature* that represents the semantic meaning of the multimedia codestream, and computes authentication data on the feature. For verification, a feature is extracted from the received codestream and is verified against the authentication data. For this reason, the problem of authenticating non-scalable codestreams in the event of transcoding can be more efficiently addressed using content-based authentication solution since a legitimate transcoding will not alter the codestream semantic.

Scalable multimedia codestreams are encoded into a multi-layered structure consisting of a mandatory base layer and one or more enhancement layers, each "enhancing" the base layer content in terms of its frame rate, resolution or SNR. Instead of encoding multiple versions of the same content to cater for different devices or network conditions, a source prepares a single scalable codestream per content

over a source-proxy-user network, where one or more proxies perform transcoding by removing one or more enhancement layers, before delivering the transcoded codestreams to end users. In this case, a carefully designed cryptographic-based authentication scheme that is compatible to the scalable codestream structure before and after transcoding is a more efficient approach. Compared to content-based authentication scheme which requires additional feature and watermark extraction phases, a cryptographic-based authentication scheme can be performed directly at the codestream level.

The contributions of this dissertation on the issue of online video codestream authentication are as follows:

- Two novel cryptographic-based authentication schemes for generic scalable codestreams modeled as a sequence of video frames, and each frame is encoded with a spatial-quality base layer and one or more spatial-quality enhancement layers are presented. The first scheme combines the advantageous features of hash-chaining and erasure-correction coding, resulting in an authentication scheme that is proxy-transparent and resilient to packet loss with low communication overhead. The second scheme further shortens the verification delay incurred at the users by using hash-based message authentication code, instead of digital signature, to protect the authenticity of individual packet. Compared to existing work, the proposed schemes simultaneously achieve lower computation costs and proxy transparency. A comprehensive analysis of both schemes in terms of their computation costs, communication overhead, buffer size requirement and loss-resiliency is also presented.

- The work for generic scalable codestreams is then extended to address the authentication for H.264/SVC codestreams, where in addition to spatial-quality layers, an H.264/SVC codestream is also encoded with a temporal base layer and one or more temporal enhancement layers. The proposed cryptographic-based authentication scheme is H.264/SVC format compliant, capable of sup-

porting all three scalabilities provided by H.264/SVC, robust to transcoding and proxy-transparent, and it incurs minimal buffer requirements at both the source and the users. The scheme is implemented on a smart phone and experiment result shows that the computation cost of the scheme is acceptable for low-power devices. To realistically assess packet loss-resiliency of the scheme, a Gilbert model which closely characterizes packet loss behavior in wireless mobile networks is employed and simulation results demonstrate that the scheme is able to achieve high verification rates (i.e., good loss-resiliency) at much lower communication overhead compared to existing schemes.

- The security of existing transform-domain content-based authentication schemes as a mean to authenticate non-scalable codestreams is studied. Existing transform-domain content-based authentication schemes are surveyed and sorted into two categories, and a common design flaw in these schemes is identified - i.e., the transform-domain feature extracted and authenticated in these schemes is insufficient to securely authenticate a codestream. As a result, although both categories of schemes are able to detect semantic-changing attacks performed in the pixel domain, they are unable to detect attacks performed in the transform domain. The ways that transform domain parameters in an authenticated codestream can be manipulated to mount semantic-changing attacks are described and several attack examples are presented. Finally, an analysis on the attacks that manipulate the transform-domain header parameters of a codestream, and the conditions of the attack given the attacker's desired attack content, is presented.

1.3.2 Access Control for Multimedia Streams

Solutions for access control fall into two categories: access control models and cryptographic-based techniques. For better scalability purpose, multimedia codestreams are commonly being stored at the proxies in a distributed fashion. For

this reason, it becomes more important to store the codestreams in encrypted form and employ cryptographic-based access control to manage accesses to protected codestreams. A cryptographic-based access control manages authorization by encrypting the codestreams such that only authorized users with the right access keys can decrypt the codestreams. This often involves the deployment of an online Key Distribution Center (KDC) to distribute access keys to authorized users.

One of the aims of this work is to design a cryptographic-based access control scheme that eliminates the usage of online KDC. An online KDC poses a scalability problem as the number of users increases and its improper management could result in a single point of failure. Secondly, the access control scheme should also be secure against collusion attacks, where two or more users subscribing to different (lower) privilege levels must not be able to collude and derive the access keys for codestreams at higher privilege levels. As multimedia codestreams are rich in internal structure, an access control scheme should employ an efficient access keys generation hierarchy that is fully compatible with the structure of the codestream and is efficient such that the source and user needs to only receive a minimal number of access keys in order to decrypt the codestream.

The contribution of this dissertation on the issue of access control for scalable multimedia codestreams is as follow:

- Two cryptographic-based access control schemes for generic and H.264/SVC codestreams, respectively, are presented. Building on top of an authentication scheme, the schemes use symmetric encryption to encrypt the codestreams and employ attribute-based encryption to disseminate access keys to authorized users. The proposed schemes employ secure and efficient key generation hierarchy that is fully compatible with the structure of the respectively codestreams. Compared to existing work, the schemes are secure against collusion attacks and are efficient in the sense that users need to only maintain a single access key regardless of their privilege levels.

1.3.3 Organization

This dissertation is organized as follow. Chapter 2 presents a comprehensive literature review on existing authentication schemes for scalable and non-scalable codestreams, as well as existing work on access control for scalable codestreams. Chapter 3 studies cryptographic-based authentication method for generic scalable codestreams and Chapter 4 extends the work to address cryptographic-based authentication for H.264/SVC codestreams. The security of existing content-based authentication schemes is studied in depth in Chapter 5 while the issue of access control is studied in Chapter 6. Finally, the dissertation conclusion and possible future work is summarized in Chapter 7.

Chapter 2

Literature Review

2.1 Cryptographic-Based Authentication for Multimedia Codestreams

Most of the existing cryptographic-based solutions for multimedia authentication are designed for non-scalable codestreams. A non-scalable codestream can be modeled as a sequence of packets. In the work of [28], the proposed scheme computes the hash of a packet, attaches the hash to its previous/next packet and the last packet in the hash chain is digitally signed. This method has a low communication overhead and computation cost but it is not resilient to packet loss; in other words, the loss of a single packet causes the hash chain to break and results in authentication failure. Subsequent works such as those in [29] [31] [54] [64] [79] [94] [103] [101] focus on devising different hash graph-based authentication method coupled with digital signature to achieve loss resiliency. In these schemes, each packet carries at least two hashes. As a result, communication overhead is a trade-off with loss-resiliency and is generally at a multiple of hashes.

Instead of embedding multiple hashes within a packet as the schemes above, the work of [61] presents the use of error correction coding as an alternative to improve loss-resiliency. In this scheme, a codestream is divided into groups of n packets, and

the hash of each packet is computed to obtain n hashes. Then, the scheme computes a digital signature on the concatenation of the hashes, and performs error correction coding on the concatenation of these hashes and the signature; the resulting codes are then dispersed to the n packets. During verification, as long as the number of packets received is at a threshold value, the hashes and the signature can be recovered to authenticate the received packets. An improvement to the work of [61] is given in [60], where the scheme in [60] further lowers the communication overhead by performing error correction coding twice on the hashes and signature. A complete and excellent survey on authentication for non-scalable multimedia codestreams can be found in [32].

For cryptographic-based authentication of scalable codestreams, the main idea is to identify the scalability structure of a codestream. In [96], three authentication schemes are designed for MPEG-4 codestreams; two of which - the Flat Authentication Scheme (FAS) and the Progressive Authentication Scheme (PAS) - use exclusive-OR and concatenation techniques respectively to generate the hash of a frame, where a frame consists of a base layer and one or more enhancement layers. It is worth noting that while these schemes are designed for MPEG-4 codestreams, they can also be applied to generic scalable codestreams. However, these schemes are not proxy-transparent, and they support only spatial-quality scalability (i.e., transcoding is performed by discarding spatial-quality layers within a frame) without considering temporal scalability. In [32], a hash-chaining technique is used for generating hashes for frames in a generic scalable multimedia codestreams. Similarly, it models a scalable codestream as a sequence of packets/frames and each frame has a base layer and multiple enhancement layers. Each enhancement layer of a frame is hashed and its hash is attached to its predecessor layer in the same frame. Then, the hash of each frame (i.e. the hash of the base layer) is appended to its previous frame, and the first frame of the group is digitally signed. This scheme is proxy-transparent but it is not loss-resilient and it also does not support temporal scalability. More recently, an authentication scheme for H.264/SVC-encoded

codestreams is presented in [56]. The scheme supports all scalabilities provided by H.264/SVC streams where temporal layers are authenticated using a hash chain. The scheme is proxy-transparent, but it has a high communication overhead.

2.2 Content-Based Authentication for Multimedia Codestreams

Content-based authentication is concerned with extracting a feature that is robust to transcoding and can truly represent the semantic meaning of the multimedia codestream. In this section, a survey is performed on existing transform-domain content-based authentication schemes, where the feature is extracted from either the transform-domain header parameters or payload (i.e., coefficients) of macroblocks (a 16×16-pixel area) in a frame.

In [18], [44], [74] and [88], a feature is extracted by computing a hash on different macroblock coefficients, which makes the schemes not robust to transcoding. In [45], [76] and [100], a feature is extracted by computing a hash of both the coefficients and a header parameter called the "motion vectors"; similarly, the schemes are also not robust to transcoding. The work of [81] extracts a feature that is robust to transcoding from the DC coefficient of macroblocks in a frame whereas the work of [102] utilizes the stable relationship between DC coefficients of two adjacent macroblocks as a robust feature. On the other hand, the work of [14] extracts a robust feature by extracting partial energy difference between coefficients of two macroblocks.

The authentication data (i.e., watermark) computed from the feature is either embedded back into the coefficients in the payload (i.e., coefficients), or into the header (i.e., prediction parameters). For embedding into payload, the rule of evaluating Least Significant Bit (LSB) [18][81][100], zero/non-zero coefficients [102] or energy relationship between coefficients [14][88] are used, whereas for embedding

into header, the rule of evaluating LSB [44][74] of motion vectors is used.

To disallow transcoding on codestreams such as those in the surveillance applications (which would lose vital details if transcoded), the work of [35] proposes to extract a feature from the header, e.g., the "directional prediction modes" of macroblocks in a frame. In [63], both the directional prediction modes and macroblock partition sizes are used as feature for authentication. These schemes are shown to reliably detect semantic-changing attacks as well as unauthorized transcoding due to the fragile nature of header parameters. The watermark is then embedded into the payload using the LSB evaluation rule.

The three most common security problems in content-based authentication schemes for images, namely undetected modifications, information leakage and protocol weakness, are summarized in [25]. For example, in [34], the authors point out that due to independent pixel-/block-wise feature and watermark extraction, the schemes in [95] and [99] are vulnerable to *collage attacks*, where an attacker can swap pixels or blocks within an image (or among database of images authenticated using the same secret key) to produce a counterfeit image. To thwart this attack, [48] proposes to extract feature from one block and embed its watermark into another randomly selected block. However, due to information leakage in watermark generation, the secret block relationship graph can be exposed by an attacker as shown in [10]. A similar flaw in [46][47] has also been exploited by [97] to expose the secret relationship graph via a *verification device attack* [25]. While there are many studies on security of content-based authentication schemes for images, not many have been done for videos, except for the work of [86], where a flaw in watermark generation process in [74] is identified. As will be discussed and shown in Chapter 5, almost all prediction parameters in the header have interdependent relationship with the payload. If such relationship cannot be fully characterized by the feature, it can be exploited to achieve undetected semantic-changing attacks.

2.3 Access Control for Multimedia Codestreams

Access control schemes for codestreams encoded by the MPEG-4 Fine Granularity Scalability (FGS) standard are proposed in [105] and [106]. In these work, it is assumed that a single encrypted MPEG-4 FGS codestream is used to cater for two different applications - one requiring scalability in terms of PSNR (peak signal-to-noise ratio) and another in terms of bit-rate. To this end, video data of an MPEG-4 FGS quality enhancement layer frame is partitioned into a group of segments. Each segment simultaneously belongs to a PSNR layer and a bit-rate layer that are correlated to each other, e.g., a low PSNR layer is likely to share data with a low (instead of a high) bit-rate layer. The security requirement in [105] and [106] is to ensure that a right to access a layer of one scalability type does not make the layers of the other scalability type also accessible, or vice versa. In [106], independent access keys are generated for encrypting each segment. This scheme is inefficient primarily due to the large number of access keys to be managed by the users. In [105], the authors exploited one-way hash function and property of the Diffie-Hellman problem to reduce the number of access keys to be managed by the users. However, the scheme is vulnerable to collusion attack and it also requires an online key distribution center (KDC) to distribute access keys to users.

In [93], an access control scheme for H.264/SVC-encoded codestreams is proposed. It treats an H.264/SVC codestream as a two dimensional structure - the *spatial-quality* scalability dimension and the *temporal* scalability dimension where the former models the spatial-quality layers of each frame and the latter models a subset of frames. Using the term "unit" to denote a portion of video data for a particular temporal and spatial-quality layer, the work of [93] encrypts each unit using an access key generated in a hierarchical manner. However, the scheme is also vulnerable to collusion attack and assumes the presence of an online KDC for access keys distribution.

Chapter 3

Cryptographic-based Authentication for Generic Scalable Codestreams

3.1 Introduction

Most of the existing cryptographic-based authentication schemes (e.g., [28], [31] and [103]) are designed for non-scalable codestreams (such as those encoded by the H.222|MPEG-2 [37] standard), where the entire codestream must be available for successful decoding. Therefore, these authentication schemes are monolithic and cannot be directly applied to scalable codestreams, because a legitimate transcoding will be viewed as tampering and the codestreams would be rejected.

A scalable multimedia codestream is encoded into a multi-layered structure. The base layer is mandatory for successful decoding and it decodes to a coarse multimedia representation. To improve the multimedia representation in terms of either frame rate, resolution or PSNR (peak signal-to-noise ratio), a decoder must further decode one or more enhancement layers in the respective scalability dimension. During transcoding, the proxy discards, starting from the highest layer, one or more enhancement layers to reduce the codestream bit-rate. Thus, apart from satisfying the security requirement, an authentication scheme designed for scalable codestreams must preserve such scalability structure in order to be robust to

transcoding.

There are only a few authentication schemes proposed for scalable multimedia codestreams. In [16], an authentication scheme for generic data modality is proposed, where sanitizable signature [5] is used to allow intermediate proxies to modify certain portion of the data without invalidating its authentication data. The scheme is then extended to address the authentication of JPEG-2000 and MPEG-4 scalable codestreams. However, due to the use of sanitizable signature, the scheme has high computation costs for the source and users, and they are not proxy-transparent. In [96], the Flat Authentication Scheme (FAS) and the Progressive Authentication Scheme (PAS) proposed for MPEG-4 video codestreams use exclusive-OR and concatenation technique, respectively, to generate the hash for a frame containing one or more layers. However, these schemes are not proxy-transparent as the proxies are required to insert auxiliary authentication information into the transcoded codestream for a successful authentication. Note that while the schemes in [96] are designed with MPEG-4 codestreams in mind, they can also be applied to authenticate generic scalable codestreams.

In the work of [32], a hash-chaining technique for authenticating generic scalable multimedia codestreams is proposed. It models a scalable codestream as a sequence of frames, where each frame has a base layer and multiple enhancement layers. The codestream is divided and processed in groups of n frames. Each enhancement layer of a frame is hashed and its hash is attached to its predecessor layer in the same frame. Then, the hash of each frame (i.e. the hash computed on the base layer) is appended to its previous frame, and the first frame of the group is digitally signed. Due to the one-way property of the hash function, the signature authenticates the entire group of frames. Note that a transcoded codestream with a base layer and zero or more enhancement layers can be verified using the authentication data carried only in that transcoded codestream. This property implies that the scheme is proxy-transparent, i.e. to transcode a multimedia codestream, a proxy simply discards certain number of enhancement layers from the top and delivers the

remaining layers to users. However, the scheme does not tolerate frame loss - once a frame is lost, all subsequent frames in a group become non-verifiable and must be rejected. The same authors also proposed an authentication scheme for H.264/SVC codestreams [56]. The scheme supports all scalabilities provided by H.264/SVC but it has a high communication overhead.

In this chapter, two schemes for authenticating generic scalable multimedia codestreams over packet-lossy source-proxy-user networks are presented. As in [32] and [96], a generic scalable multimedia codestream is modeled as a sequence of frames, where each frame contains a base layer and m $(m \geq 0)$ cumulative enhancement layers as shown in Figure 3.1. During transcoding, the proxy discards a subset of enhancement layers, starting from the highest layer. Without loss of generality, it is assumed that each network packet carries a complete frame; in the rest of this chapter, the terms "frame" and "packet" will be used interchangeably. The first scheme combines the advantageous features of [32] and [96], resulting in an authentication scheme which is proxy-transparent and resilient to packet loss with a low communication overhead. The second scheme further improves the computation cost incurred at the end user side by using Hash-based Message Authentication Code (HMAC), instead of digital signature, to protect the integrity of individual packet. Both schemes are analyzed in detail in terms of their computation costs, communication overhead, buffer size requirements, and loss-resiliency in comparison to the schemes in [32] and [96].

3.1.1 Erasure Correction Code

Let k and n be two positive integers satisfying $k < n$. An (n, k) ECC consists of an encoder module and a decoder module. The encoder module accepts a k-tuple of information symbols $X_k = (x_1, x_2, \cdots, x_k)$ and outputs an n-tuple of codeword $Y_n = (y_1, y_2, \cdots, y_n)$ where all x_i and y_j have the same bit length, $1 \leq i \leq k$ and $1 \leq j \leq n$. In an (n, k) systematic ECC, the codeword

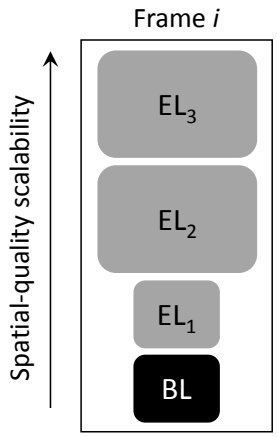


Figure 3.1: Structure of a generic scalable codestream, where each frame has an SQ (spatial-quality) base layer and $m = 3$ SQ enhancement layers. Each SQ enhancement layer cumulatively improves the base layer in terms of resolution or PSNR.

$Y_n = (x_1, x_2, \cdots, x_k, y_{k+1}, y_{k+2}, \cdots, y_n)$, where $y_{k+1}, \cdots, y_n$, are called parity check symbols [49]. Assuming that the (n, k) ECC is maximum distance separable, i.e., its minimum Hamming distance is $n - k + 1$, given any k or more symbols in Y_n, the decoder can output the original information of X_k [49].

The first authentication scheme makes use of a double-ECC coding scheme (DECS), which is similar to that used in [60], to achieve a high probability of successful authentication but with low communication overhead. A DECS scheme consists of an (n, k) systematic ECC scheme and a $(2n - k, n)$ systematic ECC scheme, denoted by $ECC_{n,k}$ and $ECC_{2n-k,n}$ respectively. The encoding and decoding functions of the DECS scheme are described below.

Encoding function DECS-EN(X_n) This function takes as input an n-tuple $X_n = (x_1, x_2, \cdots, x_n)$ where x_i is l bits long, and outputs $Z_n = (x_1 \| y_1, x_2 \| y_2, \cdots, x_n \| y_n)$, where "$\|$" denotes the concatenation of two symbols. The function proceeds in four steps:

1. Compute a $2n - k$-tuple $(x_1, x_2, \cdots, x_n, c_1, c_2, \cdots, c_{n-k}) \leftarrow ECC_{2n-k,n}(X_n)$.
2. Divide $c_1 \| c_2 \| \cdots \| c_{n-k}$ into k symbols of equal length, denoted by $(d_1, d_2 \cdots, d_k)$.

3. Compute an n-tuple $(y_1, y_2, \cdots, y_n) \leftarrow ECC_{n,k}(d_1, d_2, \cdots, d_k)$.

4. Output $W_n = (x_1 \| y_1, x_2 \| y_2, \cdots, x_n \| y_n)$.

Decoding function DECS-DE(Y_q) Suppose $Y_q = (x_{i_1} \| y_{i_1}, x_{i_2} \| y_{i_2}, \cdots, x_{i_q} \| y_{i_q})$ is a subset of W_n and $k \leq q \leq n$. The decoding function takes Y_q as input and outputs X_n with the following steps:

1. Use $ECC_{n,k}$ to decode $(y_{i_1}, y_{i_2}, \cdots, y_{i_q})$ to obtain $(d_1, d_2, \cdots, d_k)$, since $q \geq k$.

2. Divide $d_1 \| d_2 \| \cdots \| d_k$ into $n - k$ symbols of equal length, namely $(c_1, c_2, \cdots, c_{n-k})$.

3. Use $ECC_{2n-k,n}$ to decode $(x_{i_1}, x_{i_2}, \cdots, x_{i_q})$ and $(c_1, c_2, \cdots, c_{n-k})$ to get $X_n = (x_1, x_2, \cdots, x_n)$ since $q + n - k \geq n$.

The remarkable benefit of DECS is that, as long as at least k symbols in W_n are received, the DECS decoding ensures that all $(x_1, x_2, \cdots, x_n)$ will be recovered.

3.1.2 List of Notations

The list of notations used throughout this chapter is listed in Table 3.1.

Notation	Description
$\mathcal{H}(.)$	One-way hash function, e.g., SHA-1 [23]
n	Number of packets/frames in one Group-of-Frames (GOF)
m	Number of enhancement layers in a frame
t	Number of enhancement layers removed by proxy in transcoding
G_{id}, S_{id}	GOF and codestream identifier, respectively
q	Number of authenticated packets received by the user
k	Parameter for erasure correction code (ECC)
y_i, s_i	ECC codewords for packet hashes and digital signature, respectively

Table 3.1: List of notations

3.2 Signature-based Authentication Scheme

The proposed signature-based authentication scheme consists of three algorithms: the *Authentication* algorithm used by a source to generate and insert authentication data; the *Transcoding* algorithm used by a proxy to perform transcoding and the *Verification* algorithm used by an end user to verify the authenticity of received packets.

A multimedia codestream is divided into groups of n packets. Because packet groups are processed independently, the following descriptions focus on the processing of one packet group $G = [P_1, P_2, \cdots, P_n]$, where P_i denotes the i-th packet (or frame) in the group. For each packet P_i, let $L_{i,0}$ denote its base layer and $L_{i,j}$ denote its j-th enhancement layer, $j = 1, 2, \cdots, m$. Therefore, $P_i = L_{i,0} \| L_{i,1} \| \cdots, \| L_{i,j} \| \cdots \| L_{i,m}$.

During system initialization, the source chooses a digital signature scheme $\Sigma = (\mathsf{Sig}(), \mathsf{Vfy}())$ with a secret key sk and a public key pk. The source's signature on a message M is computed as $\sigma = \mathsf{Sig}_{sk}(M)$. Given signature σ and message M, anyone can verify the authenticity of M by checking whether $\mathsf{Vfy}_{pk}(\sigma, M)$ returns 1. The source's public key pk is distributed to all users in the system in an authenticated manner. The proposed scheme consists of the following three algorithms.

Authentication algorithm: Suppose that the source generates a multimedia codestream with an identifier S_{id}. Taking as input a packet group $G = [P_1, P_2, \cdots, P_n]$ with an identifier G_{id}, the source performs the following steps to output an authenticated packet group G'.

Step A1. For each packet P_i, $1 \leq i \leq n$, compute $h_{i,m} = \mathcal{H}(L_{i,m} \| m)$ as the hash of the top enhancement layer, and compute $h_{i,j} = \mathcal{H}(h_{i,j+1} \| L_{i,j} \| j)$, for layer $L_{i,j}$, $0 \leq j \leq m-1$.

Step A2. Generate the codeword $(h_{1,0} \| y_1, \cdots, h_{n,0} \| y_n)$ = $\mathsf{DECS\text{-}EN}(h_{1,0}, h_{2,0}, \cdots, h_{n,0})$.

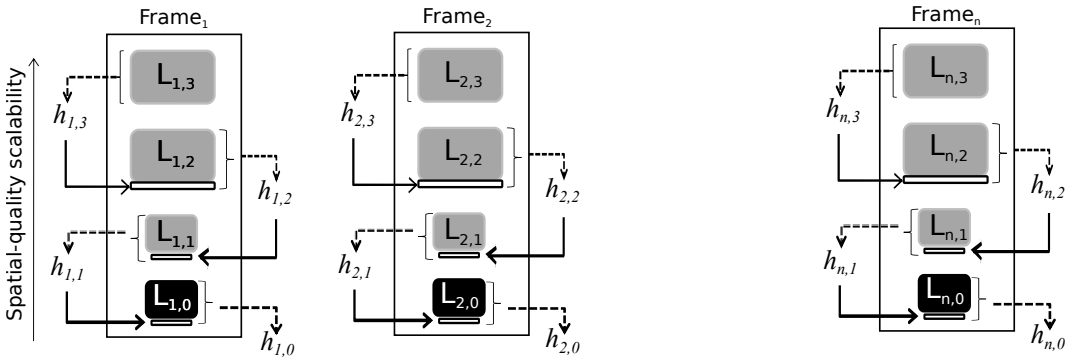


Figure 3.2: Packet hashes generation for the proposed Signature-based authentication scheme. Dotted arrow represents calculation of hash value, solid arrow indicates the action of appending.

Step A3. Compute the packet group hash $H = \mathcal{H}(h_{1,0}\|h_{2,0}\|\cdots\|h_{n,0}\|pk\|G_{id}\|S_{id})$ and $\sigma = \mathsf{Sig}_{sk}(H)$.

Step A4. Divide the binary string of σ into k equal-length segments $(\sigma_1, \sigma_2, \cdots, \sigma_k)$. Generate the codeword $(s_1, s_2, \cdots, s_n)$ by applying the $ECC_{n,k}$ encoding function on $(\sigma_1, \sigma_2, \cdots, \sigma_k)$.

Step A5. Output $G' = [P'_1, P'_2, \cdots, P'_n]$ as the authenticated packet group where $P'_1, \cdots, P'_n$ are given by

$$L'_{i,m} = L_{i,m} \tag{3.1}$$

$$L'_{i,j} = h_{i,j+1}\|L_{i,j}, \text{ for } 1 \leq j \leq m-1 \tag{3.2}$$

$$L'_{i,0} = h_{i,1}\|L_{i,0}\|y_i\|s_i \tag{3.3}$$

$$P'_i = L'_{i,0}\|L'_{i,1}\|\cdots\|L'_{i,m} \tag{3.4}$$

□

Figure 3.2 depicts the procedure for the source to generate packet hashes. In the end, the source transmits G' to the proxies.

Transcoding algorithm: Upon receiving $G' = [P'_1, \cdots, P'_n]$ from the source, the proxy transcodes it according to the downstream network bandwidth or capabilities

of user devices by removing a number of enhancement layers from the top. To remove the top t enhancement layers, $1 \leq t \leq m$, the proxy simply truncates every packet P'_i in G' into a new packet P''_i such that $P''_i = L'_{i,0} \| \cdots \| L'_{i,m-t}$. Then, the proxy forwards $G'' = [P''_1, \cdots, P''_n]$ to end users.

Verification algorithm: Due to packet loss, a user may only receive q of the n packets in G'' sent by the proxy, $k \leq q \leq n$, denoted as $\hat{G} = [P''_{i_1}, \cdots, P''_{i_q}]$, where $1 \leq i_1 < i_2 < \cdots < i_q \leq n$. If $q < k$, the user rejects $\hat{G}$ because it does not contain sufficient information for verification. Otherwise, the user runs the following steps to verify the integrity.

Step V1. For each packet $P''_j \in \hat{G}$, parse P''_j into $m - t + 1$ layers, i.e. $P''_j = L_{j,0} \| L_{j,1} \| \cdots \| L_{j,m-t}$. In addition, parse the base layer $L_{j,0}$ such that $L_{j,0} = \hat{h}_{j,1} \| \hat{L}_{j,0} \| y_j \| s_j$ and parse each enhancement layer $L_{j,l}$ such that $L_{j,l} = \hat{h}_{j,l+1} \| \hat{L}_{j,l}$ for $l \in [1, m - t]$. Namely, the user recovers $(\{\hat{L}_{j,l}\}_{l=0}^{m-t}, \{\hat{h}_{j,l}\}_{l=0}^{m-t}, y_j, s_j)$ for each packet P''_j.

Step V2. For each packet $P''_j \in \hat{G}$, compute the hash values from the $(m - t)$-th enhancement layer down to the base layer. Namely, compute $h_{j,l} = \mathcal{H}(h_{j,l+1} \| \hat{L}_{j,l} \| l)$, $0 \leq l \leq m - t - 1$ and $h_{j,m-t} = \hat{h}_{j,m-t}$.

Step V3. For all packets $P''_j \in \hat{G}$, use the hashes on their base layers and the y values to recover the hash of the packet group $\hat{G}$. Namely, compute $(h_{1,0}, h_{2,0}, \cdots, h_{n,0}) = \mathsf{DECS\text{-}DE}(h_{i_1,0} \| y_{i_1}, h_{i_2,0} \| y_{i_2}, \cdots, h_{i_q,0} \| y_{i_q})$, since $q \geq k$.

Step V4. Compute the packet group hash $H = \mathcal{H}(h_{1,0} \| h_{2,0} \| \cdots \| h_{n,0} \| pk \| G_{id} \| S_{id})$.

Step V5. Recover the signature by computing $(\sigma_1, \sigma_2, \cdots, \sigma_k) = \mathsf{DECS\text{-}DE}(s_{i_1}, s_{i_2}, \cdots, s_{i_q})$, since $q \geq k$. Set $\sigma = \sigma_1 \| \sigma_2 \| \cdots \| \sigma_k$.

Step V6. Verify the signature by checking whether $\mathsf{Vfy}_{pk}(\sigma, H)$ outputs 1. If so, accept $\hat{G}$; otherwise reject it. □

3.3 HMAC-based Authentication Scheme

It is demanding for a low-power device in the Signature-based authentication scheme to store a group of n packets and verify the digital signature without disrupting the codestream data rendition. In this section, a lightweight authentication scheme employing HMAC (hash-based message authentication code) is presented. Since the computation time of a HMAC is comparable to that of a hash computation, the scheme significantly reduces the computation cost of the end users. A prerequisite of this scheme is that the source shares a secret key k_{MAC} with all users. The key management issue will be discussed in the next section.

The HMAC-based scheme also consists of an *Authentication* algorithm for the source, a *Transcoding* algorithm for the proxies and a *Verification* algorithm for end users. The *Transcoding* algorithm is identical to that in the Signature-based scheme and hence is omitted below.

Authentication algorithm: Suppose that the source generates a multimedia codestream with an identifier S_{id}. The authentication is on the packet level. Given a packet $P = L_0\| \cdots \|L_m$, the source performs the following to output an authenticated packet $P' = L'_0\| \cdots \|L'_m$.

Step A1. Compute $h_m = \mathcal{H}(L_m\|m)$ as the hash for the top enhancement layer, and compute $h_i = \mathcal{H}(h_{i+1}\|L_i\|i)$ for all $0 < j \leq m-1$.

Step A2. Compute $h_0 = \mathsf{HMAC}(h_1\|L_0\|0\|S_{id}, k_{MAC})$.

Step A3. Set $L'_m = L_m$, $L'_i = h_{i+1}\|L_i$ for $1 \leq i \leq m-1$, and $L'_0 = h_1\|L_0\|h_0$.

Step A4. Output $P' = L'_0\|L'_1\| \cdots \|L'_m$. □

Verification algorithm: A user verifies every packet P' in a received codestream using the HMAC key k_{MAC}. Suppose that at transcoding, the proxy removes t of m enhancement layers from every packet. Let $\hat{P} = \hat{L}_0 \| \cdots \| \hat{L}_{m-t}$ be a received packet.

Step V1. For all $0 < i \leq m-t$, parse $\hat{L}_i$ into $\hat{h}_{i+1} \| L_i$ and parse $\hat{L}_0$ into $\hat{h}_1 \| L_0 \| \hat{h}_0$.

Step V2. Set $h_{m-t+1} = \hat{h}_{m-t+1}$ and compute $h_i = \mathcal{H}(h_{i+1} \| L_i \| i)$ for all $0 < i \leq m-t$.

Step V3. Compute $h_0 = \mathsf{HMAC}(h_1 \| L_0 \| 0 \| S_{id}, k_{MAC})$.

Step V4. If $h_0 = \hat{h}_0$, $\hat{P}$ is accepted and output $P = L_0 \| \cdots \| L_{m-t}$; otherwise, $\hat{P}$ is dropped. □

3.4 Security and Discussions

In the signature-based authentication scheme, the source's signature is computed over the concatenation of the hashes of all n packets where each hash carries the accumulated hashes of all enhancement layers residing in the same packet. As long as the hash function and the signature scheme are secure, the resulting authentication scheme is secure in the sense that any malicious modifications on the packets will be detected by the verification algorithm. The DECS coding scheme has no impact on security, it simply protects all the hashes so that the end user can still recover all hashes and therefore verify the signature in spite of packet loss. Similarly, the HMAC-based authentication scheme is secure as long as the underlying HMAC algorithm is secure. The security of both schemes can be proved formally using reduction [40], by showing that breaking the security of the authentication schemes leading to breaking of the hash function, signature scheme or HMAC algorithm.

The signature-based authentication scheme requires that the public key of the source be distributed to end users in a public but authenticated manner. This can be

done for example by embedding the public key in the client software. The HMAC-based authentication scheme uses secret keys for HMAC computation which normally requires the existence of a key distribution center. Another advantage of the signature-based scheme is that it provides source non-repudiation but at the expense of introducing more computational overhead due to use of signature, as will be shown in the next section.

3.5 Performance and Discussions

With reference to the source-proxy-user framework, the following metrics are defined in order to evaluate the performance of an authentication scheme designed for scalable multimedia codestreams:

- *Computation time.* The amount of time needed by the *source* as well as the *proxy* to generate authentication information for a group of n packets.
- *Verification delay:* The amount of time needed by a *user* to verify a group of q packets, $k \leq q \leq n$.
- *Per-packet authentication information (Communication overhead).* The amount of authentication information contained in a packet.
- *Buffer size.* The buffer space needed at the *source* and *user* in order to verify and process the packets.
- *Authentication probability (Loss-resiliency).* The percentage of packets that are received and verifiable.
- *Proxy transparency.* The need for proxies to be aware of the authentication mechanism.

The two proposed authentication schemes, namely the Signature- and HMAC-based schemes proposed in Sections 3.2 and 3.3 respectively, are analyzed in comparison to the FAS and PAS in [96] and the Hash-Chaining scheme in [32] since

all of them can be applied for authentication of generic scalable multimedia codestreams. As the Hash-Chaining scheme has been described in Section 3.1, the following discussion presents a brief overview of the FAS and PAS schemes.

Parameter	Value	Description
$\|\mathsf{BL}\|$	533 bytes	Average size of the base layer in a packet
$\|\mathsf{EL}\|$	133 bytes	Average size of an enhancement layer in a packet
$\|L\|$	213 bytes	Average size of a layer in a packet
m	4	The number of enhancement layers
t	2	The number of discarded enhancement layers
$\|\mathcal{H}(.)\|$, $\|\mathsf{MAC}\|$	20 bytes	Size of the output of a one-way function and a HMAC function
$\|\sigma\|$	128 bytes	Size of a digital signature
t^s_{hash}, t^s_{hmac} t^p_{hash}, t^p_{hmac}	0.5498 μs	Time taken by the source and proxy to compute a hash with 64-byte input block
t^u_{hash}, t^u_{hmac}	77.016 μs	Time taken by a user device to compute a hash with 64-byte input block
t^s_{Sig}	1.48 ms	Time taken by the source to generate a signature
t^u_{Vfy}	27.2 ms	Time taken by the user device to verify a signature
$\|H\|$	$20n + 4$	Size of packet group hash H assuming 4 bytes for group and codestream identifier G_{id} and S_{id}

Table 3.2: List of parameters

Both FAS and PAS perform authentication on a group G of n packets, each containing a base layer and m enhancement layers. For each packet P_i, the hash of layer j is computed as $h_{i,j} = \mathcal{H}(L_{i,j}\|j)$ where $L_{i,j}$ denotes the jth layer of P_i. In FAS, the packet hash $h_i = \mathcal{H}(\bigoplus_{j=0}^{m} h_{i,j})$ is the hash of the XOR of all layer hashes while in PAS, the packet hash is generated as $h_i = \mathcal{H}(\mathcal{H}(L_{i,0}\|\mathcal{H}(L_{i,1}\|\mathcal{H}(\cdots\|\mathcal{H}(L_{i,m}))))\|i)$. For both schemes, the packet group hash is computed as $H = \mathcal{H}(h_1, \cdots, h_n\|G_{id}\|S_{id})$, where G_{id} and S_{id} are the group and codestream identifiers respectively. The H is digitally signed and the n hashes along with the signature are encoded using the technique in [60].

The comparison is based on a group of n packets and the parameters used in the evaluation are shown in Table 3.2. These parameters are chosen as follow: Taking a practical example of an encoded scalable codestream from [92], where a quality

scalable sequence "Mobile" [3] at 15 frames-per-second with a base layer of 64 kbps, and $m = 4$ enhancement layers is obtained. Each consecutive enhancement layer increases the bit-rate to 80, 96, 112 and 128 kbps, respectively. Let $|\mathsf{BL}|$, $|\mathsf{EL}|$ and $|L|$ respectively denote the size of the base layer, enhancement layer and the average size of a scalable layer component in a frame (in this case, $|\mathsf{BL}|$ = 533 bytes, $|\mathsf{EL}|$ = 133 bytes and $|L| = 213$ bytes averaging over 15 frames).

In this setting, assume that the proxy removes $t = 2$ enhancement layers from the original codestream. In addition, assume that the time taken by the source and proxy to compute a hash on a 64-byte input block $t_{\mathsf{hash}}^{\{s,p\}}$ is 0.5498 μs while the time for the source to generate a signature t_{Sig}^{s} is 1.48 ms [13]. On the other hand, the time taken for a HP Hx 2790 with a 624MHz processor[1] to compute a hash on a 64-byte input block, t_{hash}^{u}, is 7.7016 μs while the time to verify a signature, t_{Vfy}^{u}, is 2.72 ms [72]. However, considering that 90-95% of the CPU processing time is used for multimedia processing, let $t_{\mathsf{hash}}^{u} = 77.016\mu$s and $t_{\mathsf{Vfy}}^{u} = 27.2$ms. In addition, let $t_{\mathsf{MAC}}^{i} = t_{\mathsf{hash}}^{i}$ for $i \in \{s, p, u\}$.

Table 3.3 shows the analytical results on the performance of the FAS, PAS, Hash-Chaining, Signature- and HMAC-based authentication schemes.

[1]This setting is chosen since most of the latest generation smartphones use processors of similar performance.

	FAS	**PAS**	**Hash-Chaining**	**Signature-based**	**HMAC-based**
Source operations when preparing n packets each with $m+1$ layers					
Computation time (ms)	$[(\frac{(m+1)\|L\|}{64}+1)\cdot n+\frac{\|H\|}{64}]\cdot t^s_{\mathsf{hash}}+t^s_{\mathsf{Sig}}$	$[(\frac{(m+1)\|L\|}{64}+\frac{m\|\mathcal{H}(.)\|}{64}+1)\cdot n+\frac{\|H\|}{64}]\cdot t^s_{\mathsf{hash}}+t^s_{\mathsf{Sig}}$	$[(\frac{(m+1)\|L\|}{64}+\frac{(m+1)\|\mathcal{H}(.)\|}{64})\cdot n]\cdot t^s_{\mathsf{hash}}+t^s_{\mathsf{Sig}}$	$[(\frac{(m+1)\|L\|}{64}+\frac{m\|\mathcal{H}(.)\|}{64})\cdot n+\frac{\|H\|}{64}]\cdot t^s_{\mathsf{hash}}+t^s_{\mathsf{Sig}}$	$[(\frac{(m+1)\|L\|}{64}+\frac{m\|\mathcal{H}(.)\|}{64})\cdot n]\cdot t^s_{\mathsf{hash}}+nt^s_{\mathsf{hmac}}$
Buffer sizeˆ	n	n	n	n	1
PPAI* (bytes)	$\frac{\|\sigma\|}{k}+(\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{k}+(\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{n}+(m+1)\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{k}+(m+\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\|\mathsf{MAC}\|+m\cdot\|\mathcal{H}(.)\|$
Proxy operations for n packets when removing t enhancement layers					
Computation time (ms)	$\frac{t\|\mathsf{EL}\|}{64}\cdot n\cdot t^p_{\mathsf{hash}}$	$(\frac{t\|\mathsf{EL}\|+(t-1)\|\mathcal{H}(.)\|}{64})\cdot n\cdot t^p_{\mathsf{hash}}$	$\emptyset$	$\emptyset$	$\emptyset$
PPAI* (bytes)	$\frac{\|\sigma\|}{k}+2(\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{k}+2(\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{n}+(m-t+2)\cdot\|\mathcal{H}(.)\|$	$\frac{\|\sigma\|}{k}+((m-t+1)+\frac{n-k}{k})\cdot\|\mathcal{H}(.)\|$	$\|\mathsf{MAC}\|+(m-t+1)\cdot\|\mathcal{H}(.)\|$
Proxy transparency	No	No	Yes	Yes	Yes
End user operations assuming it receives q packets with $(m-t+1)$ layers, $k\leq q\leq n$					
Verification delay (ms)	$[(\frac{\|\mathsf{BL}\|+(m-t)\|\mathsf{EL}\|}{64})\cdot q+q+1]\cdot t^u_{\mathsf{hash}}+t^u_{\mathsf{Vfy}}$	$[(\frac{\|\mathsf{BL}\|+(m-t)\|\mathsf{EL}\|}{64}+\frac{(m-t)\|\mathcal{H}(.)\|}{64})\cdot q+q+1]\cdot t^u_{\mathsf{hash}}+t^u_{\mathsf{Vfy}}$	$[(\frac{\|\mathsf{BL}\|+(m-t)\|\mathsf{EL}\|}{64}+\frac{(m-t+2)\|\mathcal{H}(.)\|}{64})\cdot n]\cdot t^u_{\mathsf{hash}}+t^u_{\mathsf{Vfy}}$	$[(\frac{\|\mathsf{BL}\|+(m-t)\|\mathsf{EL}\|}{64}+\frac{(m-t+1)\|\mathcal{H}(.)\|}{64}+\frac{\|H\|}{64})\cdot q]\cdot t^u_{\mathsf{hash}}+t^u_{\mathsf{Vfy}}$	$[(\frac{\|\mathsf{BL}\|+(m-t)\|\mathsf{EL}\|}{64}+\frac{(m-t+1)\|\mathcal{H}(.)\|}{64})\cdot q]\cdot t^u_{\mathsf{hash}}+qt^u_{\mathsf{hmac}}$
Buffer sizeˆ	At most n	At most n	n	At most n	At most 1

ˆ Measured in packets * Per-packet Authentication Information (communication overhead)

Table 3.3: Analytical results on performance of the FAS, PAS, Hash-Chaining, Signature- and HMAC-based authentication schemes.

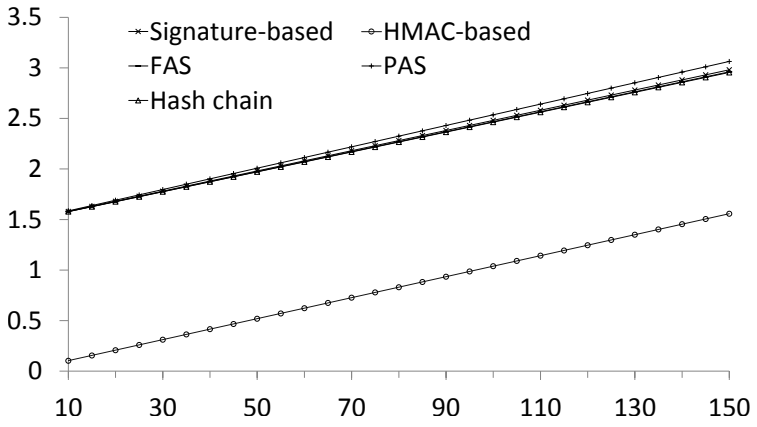


Figure 3.3: Source computation time (in milliseconds) with respect to the number of packets in a group, n, for the FAS, PAS, Hash-Chaining, Signature- and HMAC-based authentication schemes.

3.5.1 Computation time at source and proxy

Figure 3.3 shows the source computation time for the schemes under different values of n, assuming the DECS encoding/decoding overhead is negligible. Among all, PAS has the highest computation time due to the use of progressive hash followed by the Signature-based, FAS and Hash-Chaining schemes, all exhibiting almost identical computation time. A significant gap is observed in the computation time for the HMAC-based scheme against those of the other schemes. This is because the HMAC-based scheme replaces the costly signature generation with a fast HMAC computation.

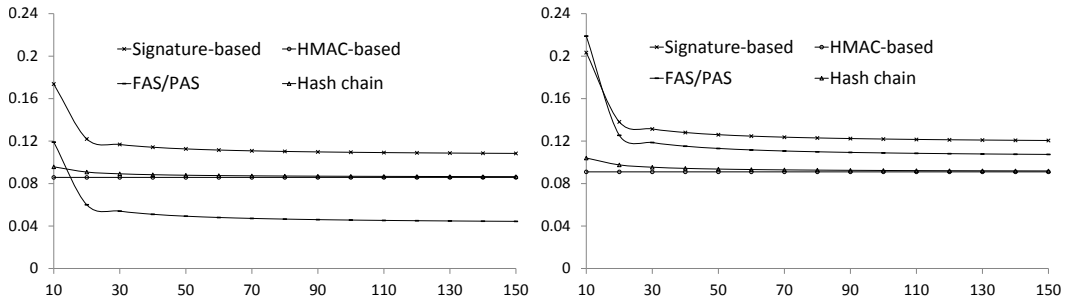
Computation time incurred at a proxy depends on whether the authentication scheme is proxy-transparent. The FAS and PAS schemes require the proxy to compute authentication information of removed layers on the fly. Hence, their computation time is linear to the number of removed layers. On the other hand, the Signature-, HMAC-based and the Hash-Chaining schemes are proxy-transparent as all necessary authentication information are already in the packets and no cost is incurred at the proxy. Among them, the Signature- and HMAC-based schemes are more desirable than the Hash-Chaining scheme as the latter does not tolerate packet loss.

3.5.2 Per-packet authentication information (communication overhead)

The FAS, PAS and Signature-based schemes employed ECC to combat packet loss. Authentication information of a group of n packets is partitioned into k symbols, ECC-coded and the resulting n symbols are dispersed across the n packets such that receiving at least k of n packets in the group will ensure recovery of the authentication information. Hence, the per-packet authentication information is inversely proportional to k and k is determined based on the packet loss probability p; if p is expected to be high, then k has to be small to accommodate more parity checks.

For a source preparing a multimedia codestream for distribution, it first estimates the anticipated packet loss probability p. Then, it selects a value α, $0 \leq \alpha \leq 1$, such that with $k = \alpha n$ data packets, an acceptable (as determined by the source) authentication probability under loss probability p can be achieved. With the values of k and n, the per-packet authentication information can then be calculated as shown in Table 3.3.

As such, before transcoding, for the FAS, PAS and Signature-based schemes, the per-packet authentication information increases linearly with p while for the Hash-Chaining and HMAC-based schemes, it remains constant regardless of p because each packet can be individually authenticated. Figure 3.4 shows the per-packet authentication overhead as a ratio (i. e., authentication data over packet length) at $p = 0.5$ and $k = 0.3n$ so as to achieve an authentication probability of ≥ 0.97. As shown in Figure 3.4(a), before transcoding, the Signature-based scheme has the highest ratio, followed by the Hash-Chaining, HMAC-based, FAS and PAS schemes. FAS and PAS have the lowest ratio because each packet in the three other schemes incorporated authentication information of all layers within the packet. However, it is worth noting that after transcoding, although the ratio of the Signature-based scheme remains the highest, it is now followed by FAS and PAS, then the Hash-Chaining and HMAC-based schemes as shown in Figure 3.4(b). A



(a) Before transcoding (b) After transcoding

Figure 3.4: Per-packet authentication information (in ratio) vs. number of packets in a group, n, for the Signature-, HMAC-based, Hash-Chaining, FAS and PAS schemes, where $p = 0.5$ and $k = 0.3n$.

more significant increase (6.3%) is observed in the ratio of FAS and PAS compared to the others. Such increase is due to the reason that after transcoding, additional authentication information of the removed layers needed to be incorporated into the packets of FAS and PAS schemes.

3.5.3 Verification delay at the user

Similar to the per-packet authentication overhead, the verification delay at the end user side is also indirectly affected by the parameter k, i.e. the number of received packets q where $k \leq q \leq n$. Intuitively, the verification delay will be longer when p is low. Figure 3.5 compares the verification delay of the schemes with respect to n for $p = 0.5$ and $k = 0.3n$. In this figure, the Hash-Chaining scheme has the longest verification delay because all of the n packets have to be received and verified in order to authenticate a packet group. It is then followed by the Signature-based, FAS and PAS schemes having almost identical verification delays. The HMAC-based authentication scheme has a constant verification delay equivalent to that of verifying a single packet because it allows user to check the integrity of individual packets, rather than an entire packet group.

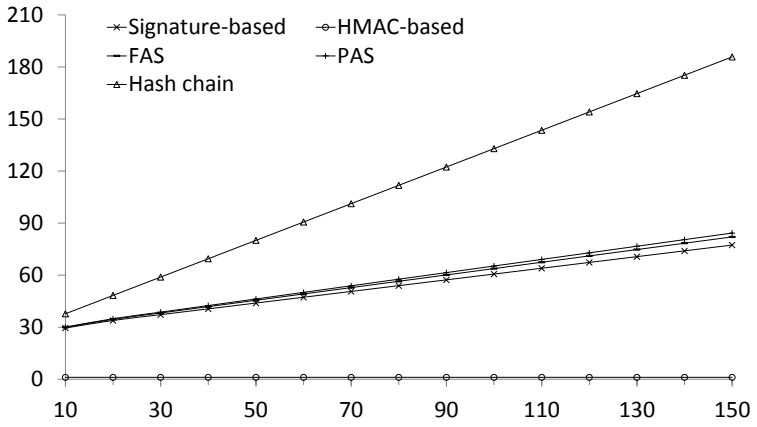


Figure 3.5: Verification delays (in milliseconds) with respect to the number of packets in a group n for the Signature-, HMAC-based authentication scheme, FAS, PAS and Hash-Chaining scheme ($p = 0.5, k = 0.3n$).

3.5.4 Authentication probability (loss-resiliency)

In the proposed Signature-based authentication scheme, a packet group can be verified successfully only if the number of lost packets in a group is $(n - k)$ or less. Assuming independent packet loss, the probability of successful authentication can be modeled as

$$P_{\text{Success}} = \sum_{i=0}^{n-k} \binom{n}{i} p^k (1-p)^{n-i}$$

where p is the loss probability. Figure 3.6 shows the authentication probability respective to different per-packet authentication information. With approximately 115 bytes of per-packet authentication information ($\approx 0.09\%$ of the packet size), the Signature-based scheme can achieve a 99% authentication probability when $p = 0.5$.

From the above results, the following can be concluded:

- The Hash-Chaining scheme is not suitable for multimedia distribution over packet-lossy networks because it cannot tolerate packet loss while the HMAC-based scheme is the most robust against network loss as each packet is individually verifiable.

- For applications which are not sensitive to delays, the Signature-based, FAS or PAS schemes can be used. For applications requiring minimal delay, the

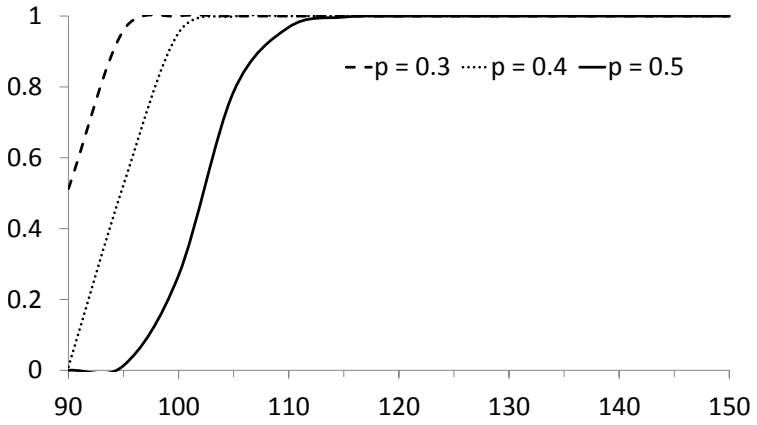


Figure 3.6: Authentication probability vs. per-packet authentication information (bytes) for the Signature-based scheme under different packet loss probabilities p (assuming $n = 128$ packets).

HMAC-based scheme outperforms the others.

- Although the FAS and PAS have the lowest initial authentication ratio, they exhibit similar performance as the Signature- and HMAC-based schemes after transcoding (note that limited bandwidth often occurs in the proxy-user communication link). Hence, the Signature- and HMAC-based schemes are better choices since they are proxy-transparent while the FAS and PAS are not.

- For applications requiring non-repudiation of origin service, only schemes using digital signatures, such as the Signature-based, FAS and PAS schemes, can be used. Though the HMAC-based scheme outperforms all the other schemes, it only provides codestream authentication, not non-repudiation, service.

The authentication schemes proposed in [16], [32] and [29] are based on the Merkle hash tree. In these schemes, a leaf node is the hash of a layer in a frame, an interior node is the hash of concatenation of its children, and the root represents the hash of the frame. This set of roots then form the leaves of another hash tree, the root of which represents the hash of a group of frames. In these schemes, the source needs at least $2^{\lceil \log_2(n(m+1))+1 \rceil} - 1$ hash computations for tree construction

assuming n frames in a group, each frame having $(m + 1)$ layers. Furthermore, the schemes are not proxy-transparent as the proxy needs to incorporate the hash of the removed layer and/or hashes of subtrees covering the removed layers to the packets. To provide proxy transparency, each leaf node needs to carry a signature generated on the root and hashes of the siblings nodes on the path to the root node ($\approx \log_2(m+1) \cdot S_{hash}$). This incurs a high computation and communication overhead.

3.6 Discussions

Two authentication schemes for generic scalable multimedia codestreams are proposed. The first scheme, namely the Signature-based authentication scheme, uses a hash chain to model the scalability structure of layers within a frame coupled with a digital signature to authenticate a group of frames. A double error correction coding scheme is used to provide loss resiliency with a low communication overhead. The second scheme, namely the HMAC-based authentication scheme, replaces the signature with a HMAC and offers packet-level authentication with a lower computation cost and communication overhead. Both schemes have the salient features of proxy-transparency with lower computation times and verification delays (for the source and users respectively) compared to the existing schemes that are not proxy-transparent. The HMAC-based authentication scheme has the lowest computation and verification costs and the strongest resilience to packet loss, which makes it highly suitable for low-end user devices with limited computation power and noisy network access (such as wireless or mobile networks).

Chapter 4

Cryptographic-based Authentication for H.264/SVC

4.1 Introduction

The H.264/SVC multimedia coding standard encodes a video codestream into 3-dimensional scalability using hierarchical prediction and inter-layer prediction structures. The hierarchical prediction structure provides temporal scalability; it processes the frames within a Group-of-Frames (GOF) and groups them into several temporal layers such that a frame in a higher temporal layer uses the preceding and succeeding frames in the lower temporal layers as reference during decoding, as shown in Figure 4.1. The frame in the lowest temporal layer, i.e., temporal base layer, is intra-predicted (encoded by exploiting similarities within the frame itself) while the frames in the enhancement layers are inter-predicted (encoded by exploiting similarities between adjacent frames). Temporal scalability is achieved by discarding all frames in a higher temporal enhancement layer, which will reduce the frame rate by half. Due to this structure, a GOF with T temporal layers consists of 2^{T-1} frames. In Figure 4.1, TL_1 is the temporal base layer while $\mathsf{TL}_2, \mathsf{TL}_3, \mathsf{TL}_4$ are the temporal enhancement layers.

Inter-layer prediction, which maximizes usage of lower layer information to im-

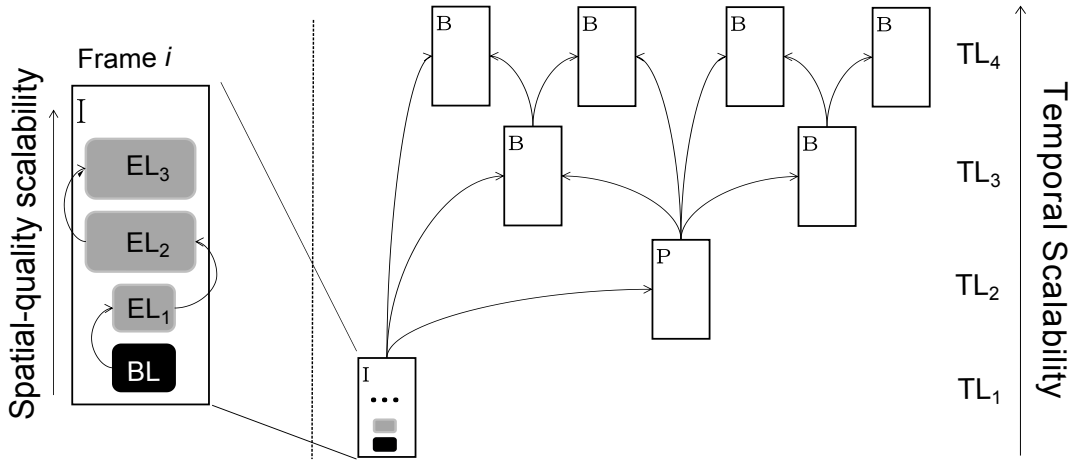


Figure 4.1: On the left - Inter-layer prediction structure within an frame; On the right - Hierarchical prediction structure within a GOF. Arrow pointing from A to B indicates that A is the reference of B during decoding.

prove the rate-distortion efficiency of a higher layer, provides spatial (resolution) and quality (PSNR, peak signal-to-noise ratio) scalabilities within a frame; it produces a base layer with the lowest resolution and PSNR, and multiple enhancement layers which improve the resolution and/or PSNR. Note that each spatial layer may have one or more quality layers and a higher spatial/quality layer utilizes a lower spatial/quality layer as reference as shown on the left hand side of Figure 4.1. As such, they are considered as a single dimension and are referred to as "SQ (spatial-quality) scalability"; a frame can be transcoded to a lower spatial/quality frame by discarding higher SQ layers.

Before an H.264/SVC codestream is transported over the network, it is organized into Network Abstraction Layer units (NALU). NALUs are designed to form natural packet boundaries and can easily handle codestream transcoding. NALUs are categorized into Video Coding Layer (VCL) NALUs and non-VCL NALUs. The VCL NALUs carry coded video data while non-VCL NALUs carry associated additional information such as SVC header information that may assist certain system operations. In the non-VCL NALUs category, the Supplemental Enhancement Information (SEI) NALU (which is a non-VCL NALU that provides additional information to assist the decoding or codestream manipulation process) that is in-

cluded in a Payload Content Scalability Information (PACSI) NALU [90] is used to achieve format-compliance for the proposed authentication scheme.

In the literature, the H.264/SVC codestream authentication scheme proposed in [56] is the first cryptographic-based authentication scheme that preserves all three dimensional scalabilities of an H.264/SVC codestream. In this scheme, the spatial and quality layers within a frame are authenticated using a directed acyclic graph and the temporal layers are authenticated using a hash chain. Authentication data is protected using ECC (Erasure Correction Code) and packet replications. The scheme is proxy-transparent and is shown to be resilient to bursty packet losses using packet interleaving technique, but it incurs a relatively high communication overhead and requires larger buffer size since it processes several GOFs (Group-of-Frames) in one authentication. Similarly, there is also a longer source computation time since the source's signature is generated over several GOFs, which makes it unsuitable for real-time applications. In addition, the use of hash chain on temporal layers makes the scheme less robust to packet losses (as will be discussed in Section 4.2).

In this chapter, an improved codestream authentication scheme focusing on the H.264/SVC Network Abstraction Layer Units (NALUs) is proposed. The proposed scheme seamlessly integrates cryptographic algorithms and erasure correction code to H.264/SVC codestreams such that the authenticated codestreams are format compliant with the H.264 specifications and preserves the 3-dimensional scalability of the original codestreams. Specifically, the proposed scheme is proxy-transparent, robust to transcoding and highly loss-resilient. To realistically assess packet loss-resiliency of the scheme, a Gilbert model [30] which closely characterizes packet loss behavior in wireless mobile networks is utilized. The scheme is implemented on a smart phone and its performance (e.g., computation costs, communication overhead and loss resiliency) over a bursty network is studied and compared to the scheme in [56].

4.1.1 List of Notations

The list of notations used throughout this chapter is listed in Table 4.1.

Notation	Description
$\mathcal{H}(.)$	One-way hash function, e.g., SHA-1 [23]
n	Number of packets/frames in one Group-of-Frames (GOF)
m	Number of spatial-quality enhancement layers in a frame
T	Number of temporal layers in a codestream
t	Number of enhancement layers removed by proxy in transcoding
G_{id}, S_{id}	GOF and codestream identifier, respectively
q	Number of authenticated packets received by the user
k	Parameter for erasure correction code (ECC)
y_i, s_i	ECC codewords for packet hashes and signature, respectively

Table 4.1: List of notations

4.2 Authentication of H.264/SVC Codestreams

Assuming that a GOF G has T temporal layers, where TL_1 is the temporal base layer (BL) and TL_l ($l \in [2, T]$) is the l^{th} temporal enhancement layer (EL), thus having a total of 2^{T-1} frames, i.e., $G = [\mathsf{F}_1, \mathsf{F}_2, \cdots, \mathsf{F}_{2^{T-1}}]$. For the spatial-quality (SQ) scalability, assume that each frame has $m + 1$ SQ layers, then, $\mathsf{F}_i = \{\mathsf{Pre}_i, \mathsf{BL}_i, \mathsf{EL}_{i,1}, \mathsf{EL}_{i,2}, \cdots, \mathsf{EL}_{i,m}\}$, where Pre_i is a 4- or 5-byte non-VCL Prefix NALU carrying the (spatial, quality, temporal) ID of BL_i; BL_i is the SQ BL NALU and $\mathsf{EL}_{i,j}$ is the j^{th} SQ EL NALU of F_i. The proposed scheme exploits the advantages of [32] and [60] and is tailored explicitly for authenticating H.264/SVC codestreams.

In the proposed scheme, an H.264/SVC codestream is processed in GOFs, each with $n = 2^{T-1}$ frames. For each F_i, a hash chain is formed starting from the highest SQ EL NALU as in [32] and its final hash (hash of BL_i), denoted h_i, is regarded as the frame hash. Note that this hash chain approach makes the scheme proxy-transparent since SQ scalability is achieved by discarding SQ NALUs starting from

the highest SQ EL NALU. Next, the hash H_l of the temporal layer TL_l is obtained by concatenating hashes of all the frames in a temporal layer. A GOF hash is then computed over hashes of all temporal layers and a source signature is generated on the GOF hash. All the hashes are finally encapsulated into SEI NALUs.

To protect against authentication data loss, a combination of DECS (reviewed in Chapter 3) and packet replications is employed. Unlike the scheme in [56] which uses packet interleaving to convert burst loss to random loss pattern, no interleaving is used in the proposed scheme in order to reduce decoding delays and to save buffer space. The proposed scheme also overcomes a limitation in [56], where if ECC is unable to recover frame hashes in temporal layer TL_l, although the frames in $\mathsf{TL}_{l+1}, \cdots, \mathsf{TL}_T$ are received correctly and usable for decoding using error concealment technique, they are unverifiable and must be discarded. In addition, as will be shown later, the proposed scheme has a much lower communication overhead and does not adversely affect the PSNR of the underlying codestream.

During system initialization, the source chooses a digital signature scheme $\Sigma = (\mathsf{Sig}(), \mathsf{Vfy}())$ with a secret key sk and a public key pk. The source's signature on a message M is computed as $\sigma = \mathsf{Sig}_{sk}(M)$. Given signature σ and message M, anyone can verify the authenticity of M by checking whether $\mathsf{Vfy}_{pk}(\sigma, M)$ returns 1. The source's public key pk is distributed to all entities in the system in an authenticated manner. In the following, the source takes as input a GOF $G = [\mathsf{F}_1, \mathsf{F}_2, \cdots, \mathsf{F}_{2^{T-1}}]$ and outputs an authenticated GOF supporting SQ and temporal scalability.

Authentication algorithm for spatial and quality (SQ) layers: To authenticate the SQ NALUs within an frame, the source takes as input a frame $\mathsf{F}_i = \{\mathsf{Pre}_i, \mathsf{BL}_i, \mathsf{EL}_{i,1}, \cdots, \mathsf{EL}_{i,m}\}$ and produces an authenticated frame F'_i (see Figure 4.2).

Step A1. Let $h_{i,m} = \mathcal{H}(\mathsf{EL}_{i,m} \| m); h_{i,j} = \mathcal{H}(\mathsf{EL}_{i,j} \| j \| h_{i,j+1}),\ j \in [1, m-1]$.

Step A2. Output $\mathsf{F}'_i = \mathsf{F}_i \bigcup \{h_{i,j}\}_{j=1}^{m}$.

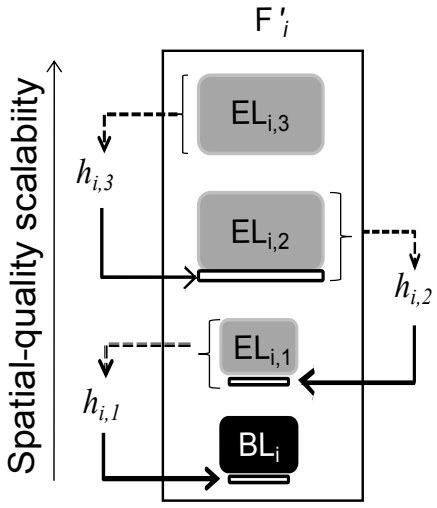


Figure 4.2: Authentication of spatial-quality layers in an H.264/SVC frame. Dotted arrow represents calculation of hash value, solid arrow indicates the action of appending.

As mentioned earlier, the hash $h_{i,j}$ is put into an SEI NALU with the same (spatial, quality, temporal)-ID as $\mathsf{EL}_{i,j-1}$ for format compliance and transparent transcoding purposes. Note that the SQ dependency structure may sometimes be a directed acyclic graph depending on encoding configurations. In this case, a similar approach as in [56] can be employed. Thus, some NALUs may carry several hashes, but the total number of hashes within an frame remains the same.

Authentication algorithm for temporal layers and GOF: At this stage, the source has a group G' containing the set of frames $[\mathsf{F}'_1, \mathsf{F}'_2, \cdots, \mathsf{F}'_{2^{T-1}}]$. In the following, the source further operates on the frames within G' to support temporal scalability.

Step A3. For each F'_i in G', compute the hash of F'_i as $h_i = \mathcal{H}(\mathsf{BL}_i \| i \| h_{i,1})$.

Step A4. Let n_l be the number of frames in temporal layer TL_l, then $\mathsf{TL}_l = [\mathsf{F}'_{i_1}, \mathsf{F}'_{i_2}, \cdots, \mathsf{F}'_{i_{n_l}}]$[1]. For each temporal layer TL_l ($l \in [1, T]$), compute the temporal layer hash as $H_l = \mathcal{H}(h_{i_1} \| h_{i_2} \| \cdots \| h_{i_{n_l}} \| l)$.

[1]Note that $n_1 + n_2 + \cdots + n_T = 2^{T-1}$.

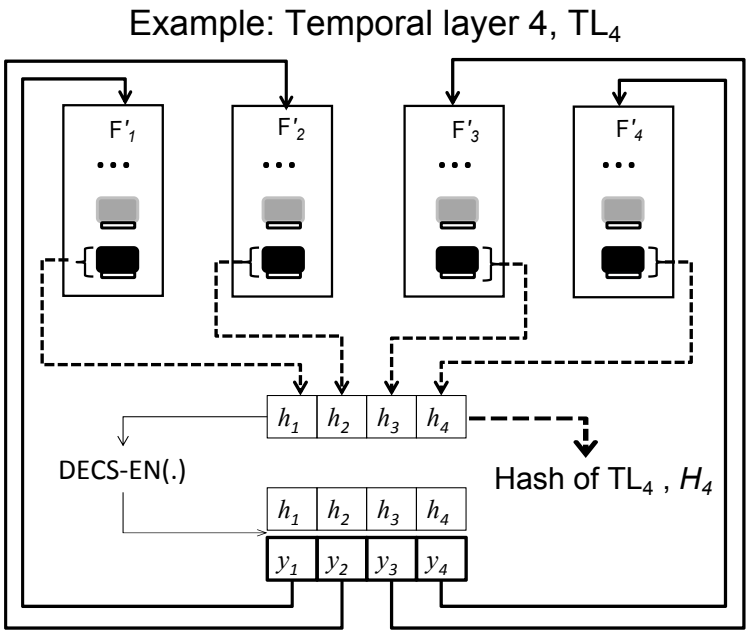


Figure 4.3: Generation and placement of DECS-EN codewords for frame hashes within a temporal layer. Dotted arrow represents calculation of hash value, **bold** solid arrow indicates the action of appending.

Step A5. Compute the GOF hash as $H_{GOF} = \mathcal{H}(H_1 \| H_2 \| \cdots \| H_T \| pk \| G_{id} \| S_{id})$, where G_{id} is the GOF identifier and S_{id} is the codestream identifier. Compute $\sigma = \mathsf{Sig}_{sk}(H_{GOF})$.

Step A6. For each temporal layer TL_l ($l \in [1, T]$), generate the codeword:

$$(h_{i_1} \| y_{i_1}, h_{i_2} \| y_{i_2}, \cdots, h_{i_{n_l}} \| y_{i_{n_l}}) \leftarrow \mathsf{DECS\text{-}EN}_{n_l,k_l}(h_{i_1}, h_{i_2}, \cdots, h_{i_{n_l}})$$

and let $\mathsf{F}'_{i_b} \leftarrow \mathsf{F}'_{i_b} \bigcup y_{i_b}$ $(1 \leq b \leq n_l)$. Therefore, in TL_l, receiving any k_l SQ BL NALUs and the DECS code symbols y_{i_b} allows the user to recover the hashes of all frames in TL_l and hence reconstruct H_l (see Figure 4.3).

Step A7. Let $S = H_1 \| H_2 \| \cdots \| H_T \| \sigma$. Output the authenticated GOF as $G' = [\mathsf{F}'_1, \mathsf{F}'_2, \cdots, \mathsf{F}'_{2^{T-1}}] \bigcup S$, where S is also placed in an SEI NALU as in Figure 4.4.

In essence, the source generates a signature over all temporal layer hashes in a GOF, each temporal layer hash is computed over the hashes of all its frames, and

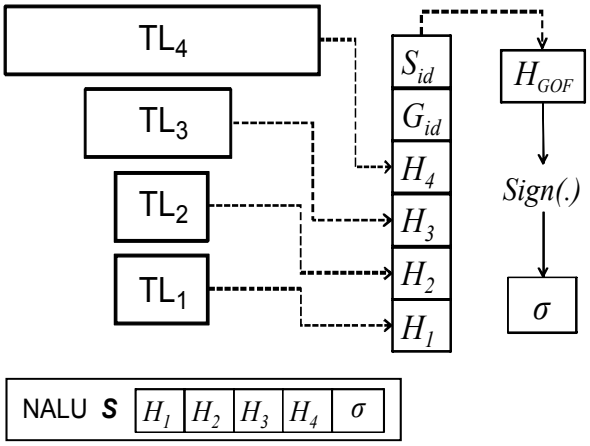


Figure 4.4: Authentication of temporal layers and GOF, and the formation of NALU S; dotted arrow indicates the calculation of hash value.

the hash of a frame carries the accumulated hashes of its SQ NALUs.

The source uses DECS in each temporal layer to protect the frame hashes. If a user wishes to obtain a particular temporal layer but some of its frames are lost, then DECS can recover the lost frame hashes; otherwise, the user simply discards all the received frames in that temporal layer and proceeds to verify frames in all other temporal layers. Note that this cannot be achieved by the scheme in [56] because temporal layers are authenticated using a hash chain. In [56], the n_T frame hashes in TL_T are concatenated, segmented into k_{T-1} pieces, ECC encoded into n_{T-1} codewords and appended to the n_{T-1} frames in TL_{T-1}. For each frame in $\mathsf{TL}_l,\ l \in [1, T-1]$, the frame hash is then computed from the codeword it carries, along with its SQ NALU. The frame hash in TL_1 carries the accumulated hashes of higher layer frames and it is regarded as the GOF hash. As a result, in any temporal layer TL_l, if less than k_l ECC codewords are received, the frames hashes in TL_{l+1} cannot be recovered and consequently, all frames in the higher temporal layers must be discarded. In addition, due to the hierarchical prediction structure, $n_{l-1} = \frac{1}{2}n_l$, ECC coding n_l hashes such that receiving k_{l-1} $(k_{l-1} < n_{l-1})$ can recover all n_l hashes resulted in a very high communication overhead. On the other hand, the use of DECS within a temporal layer incurs a much lower communication overhead in the proposed scheme.

Note that for a codestream with a large number of temporal layers (e.g., $T > 20$), the computation costs for both the source and user may become non-negligible. In this case, the authentication scheme could be modified to selectively combine one or more temporal layers and compute a single hash (and DECS) for the temporal layers. This will improve the scheme efficiency but at the expense of compromising the degree of scalability of the original codestream. To this end, note that as the number of temporal layers increases, so does the GOF size, i.e., the number of frames in a GOF is equal to 2^{T-1}, where T is the number of temporal layers. Although a larger GOF size will improve the codestream quality, but it also result in a longer decoding delay and a lower resiliency to loss as the gap between two intra-predicted frames becomes larger. Hence, the GOF size is commonly restricted to a small value and the best coding efficiency is achieved for GOF sizes between 8 and 32, i.e., $T = 4$ and 6, respectively [75][82]. Furthermore, current standard definition videos are encoded at either 25 or 30 frames per second whereas the latest ultra high definition television can support up to 60 frames per second [2][21][22]. As GOF size is restricted to be equal or less than the intended frame rate, the number of temporal layers is also restricted to $4 \leq T \leq 6$.

Because the authentication data S in Step A7 carries the signature, it is important for the SEI NALU carrying S be received, thus, to increase the probability of it being received, several replications of this SEI NALU is transmitted. In Section 4.3, it is shown that by transmitting three copies of this NALU, a near 98% verification rate in typical wireless mobile networks can be achieved.

Transcoding algorithm: Upon receiving an authenticated GOF $G' = [\mathsf{F}'_1, \mathsf{F}'_2, \cdots, \mathsf{F}'_{2^{T-1}}]$ from the source, a proxy performs transcoding operation according to network bandwidth or user device capabilities by discarding some NALUs, starting from the highest SQ EL NALUs and/or the frames in the highest temporal layer.

Suppose that the proxy desires to remove t $(t < m)$ SQ EL NALUs and

s ($s < T$) temporal layers, it first discards frames in temporal layers $\mathsf{TL}_T, \mathsf{TL}_{T-1}$, $\cdots$, TL_{T-s+1}. Let $T' = T - s - 1$, then $G' = [\mathsf{F}'_1, \mathsf{F}'_2, \cdots, \mathsf{F}'_{2^{T'}}]$. For each F'_i ($i \in [1, 2^{T'}]$), the proxy discards $\mathsf{EL}'_{i,m}, \mathsf{EL}'_{i,m-1}, \cdots, \mathsf{EL}'_{i,m-t+1}$ to form the new $\mathsf{F}''_i = \{y_i, \mathsf{Pre}_i, \mathsf{BL}_i, h_{i,1}, \mathsf{EL}_{i,1}, h_{i,2}, \mathsf{EL}_{i,2}, \cdots, h_{i,m'}, \mathsf{EL}_{i,m'}, h_{i,m'+1}\}$, where $m' = m - t$. Finally, the proxy also discards the corresponding SEI NALUs and transmits the transcoded GOF $G'' = [\mathsf{F}''_1, \mathsf{F}''_2, \cdots, \mathsf{F}''_{2^{T'}}]$ to end users. Note that in this process, the proxy simply discards NALUs and/or frames without needing to understand the authentication mechanism.

Verification algorithm: A user performs the following verification steps as she receives a GOF $\tilde{G} = [\tilde{\mathsf{F}}_1, \tilde{\mathsf{F}}_2, \cdots, \tilde{\mathsf{F}}_{2^{T'}}]$.

Step V1. If the SEI NALU containing $\tilde{S}$ is received, go to Step 2. Otherwise, $\tilde{G}$ is not verifiable and is discarded.

Step V2. Parse $\tilde{\mathsf{F}}_i$ into $\{y_i, \mathsf{Pre}_i, \mathsf{BL}_i, h_{i,1}, \mathsf{EL}_{i,1}, \cdots, h_{i,m'}, \mathsf{EL}_{i,m'}\}$ if $m' = m$, or into $\{y_i, \mathsf{Pre}_i, \mathsf{BL}_i, h_{i,1}, \mathsf{EL}_{i,1}, \cdots, h_{i,m'}, \mathsf{EL}_{i,m'}, h_{i,m'+1}\}$ if $m' < m$.

Step V3. For each $\tilde{\mathsf{F}}_i$, compute its hash starting from the m'^{th} SQ EL NALU, i.e. $\tilde{h}_{i,m'} = \mathcal{H}(\mathsf{EL}_{i,m'} \| m')$ if $m' = m$, or $\tilde{h}_{i,m'} = \mathcal{H}(\mathsf{EL}_{i,m'} \| m' \| h_{i,m'+1})$ if $m' < m$, and in other cases, $\tilde{h}_{i,j} = \mathcal{H}(\mathsf{EL}_{i,j} \| j \| \tilde{h}_{i,j+1})$; finally, $\tilde{h}_i = \mathcal{H}(\mathsf{BL}_i \| i \| \tilde{h}_{i,1})$.

Step V4. Group the frames based on the temporal layer ID, compute the temporal layer hash as $\tilde{H}_l = \mathcal{H}(\tilde{h}_{i_1} \| \tilde{h}_{i_2} \| \cdots \| \tilde{h}_{i_{n_l}} \| l)$. In the case of packet loss, as long as k_l or more SQ BL NALUs and SEI NALUs are received, the user performs $\mathsf{DECS\text{-}DE}$ to recover the missing frame hashes. If less than k_l SQ BL NALUs and SEI NALUs are received, discard the remaining received NALUs.

Step V5. Compute the GOF hash as $\tilde{H}_{GOF} = \mathcal{H}(\tilde{H}_1 \| \tilde{H}_2 \| \cdots \| \tilde{H}_T \| pk \| G_{id} \| S_{id})$ where $\tilde{H}_{T'+1}, \cdots, \tilde{H}_T$ are recovered from $\tilde{S}$.

Step V6. Verify the signature σ against $\tilde{H}_{GOF}$. If the verification outputs true, i.e., $1 \leftarrow \mathsf{Vfy}_{pk}(\sigma, \tilde{H}_{GOF})$, accept $\tilde{G}$; otherwise, reject it.

Forbidden Sequence	Replacement
0×000000	0×00000300
0×000001	0×00000301
0×000002	0×00000302
0×000003	0×00000303

Table 4.2: Forbidden sequences and their replacements.

4.3 Performance Evaluation

The proposed scheme is implemented and integrated into an open source SVC decoder (Opensvc [27]) on the Android V1.5 platform. In addition, an open source ECC [65] and signature software [59] which, respectively, implement the Reed-Solomon Code [68] and the RSA signature [73] with the hash function SHA-1 [23] are utilized. These C source codes are cross-compiled on the ARM-based smart phone on a X86-based desktop. The binary code is encapsulated in JNI (JAVA Native Interface) and JAVA renders the User Interface.

In the implementation, a slight modification on the proposed scheme is performed as follow. Firstly, note that 0x00000001 and 0x000001 are forbidden sequences in the H.264/SVC codestream because they mark the NALU boundary [84]. If such sequences occur in authentication data, they are replaced as shown in Table 4.2 and recovered as the NALU is read from the codestream. Secondly, knowledge of the position of each frame in a GOF is necessary for DECS to recover the frame hashes in the event of packet loss. Such position information is inserted into the SEI NALU corresponding to the SQ BL NALU.

4.3.1 Loss-Resiliency and Communication Overhead

A multimedia codestream can tolerate loss at the cost of a lower PSNR, but an authenticated codestream has a smaller window of tolerance because all data in a GOF must be discarded if it cannot be verified. In the following, loss-resiliency of the proposed scheme is studied by measuring its *verification rate*, defined as the ratio of

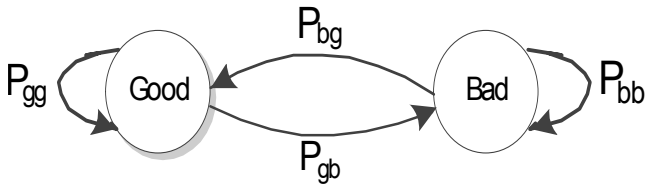


Figure 4.5: Gilbert channel model

	Scenario 1	**Scenario 2**	**Scenario 3**	**Scenario 4**
Bit-rate (kbps)	500	500	2000	2000
MTU (bytes)	300	1200	300	1200
P_{gb}	7.43e-04	2.90e-03	2.50e-04	7.30e-04
P_{bg}	4.92e-03	2.89e-02	2.49e-03	6.79e-03
P_{loss}	1.31e-01	0.91e-01	0.91e-01	0.97e-01
β	204	34	402	148
$\%$ loss	97%	65%	48%	71%

Table 4.3: Gilbert model parameters.

"the number of verifiable NALUs" to "the number of received NALUs", under the Gilbert model [30] (see Figure 4.5). In contrast to the independent loss model, the Gilbert model is a more realistic approximation to real wireless mobile networks. In this model, the two states have different packet loss probabilities. The probabilities P_{gg}, P_{gb}, P_{bg}, P_{bb} respectively represents the state transition probabilities of "Good-to-Good", "Good-to-Bad", "Bad-to-Good" and "Bad-to-Bad".

In [24], the authors formulated a packet loss model aimed at H.264 video transmission over the IEEE 802.11g Wireless LANs. This model assumes no packet loss in "Good" state and some packet loss in "Bad" state. In the following experiments, it is assumed that packet is always lost in the "Bad" state and the measurement data in [24] is adapted as listed in Table 4.3). The four scenarios in Table 4.3 respectively depicts a wireless transmission for low (500 kbps) and high (2000 kbps) bit-rate codestreams under a relatively small (300 bytes) and large (1200 bytes) Maximum Transmission Unit (MTU) size.

The average packet loss probability P_{loss} increases with burst loss length β (in packets), packet rate and MTU size. From Table 4.3, Scenario 1 has the highest

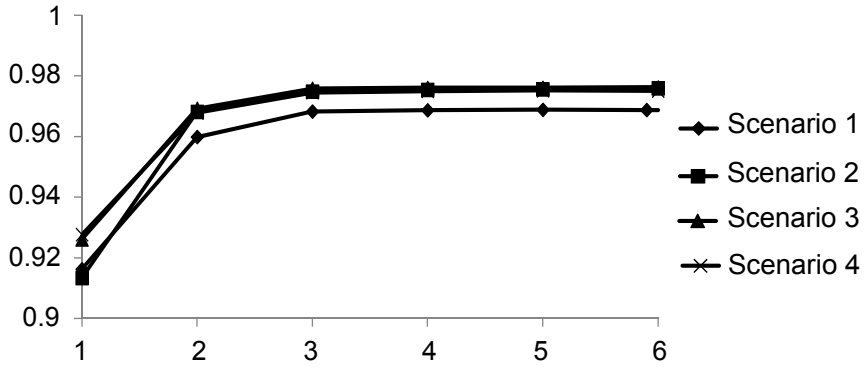


Figure 4.6: Probability of receiving NALU S vs. number of copies of NALU S.

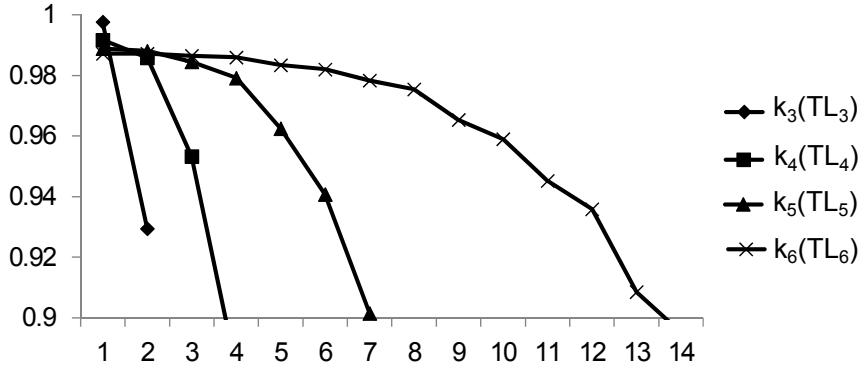


Figure 4.7: Probability of temporal layer authentication vs. its DECS parameter.

P_{loss} while Scenarios 2, 3 and 4 exhibit similar P_{loss}. This is because Scenario 1 has a larger β and a higher packet rate compared to Scenarios 2 and 4, while between the latter two, Scenario 4 has a larger β, packet rate as well as MTU size. The lower P_{loss} in Scenario 3 could be attributed to its small MTU size, however, its high packet rate results in the largest β. A large β in a multimedia codestream causes a lower verification rate for low bit-rate codestreams compared to high bit-rate codestreams because of a larger percentage of video/authentication data that is lost (% loss) in one burst. Thus, for low bit-rate codestreams, the verification rate will be lower in Scenario 1 compared to Scenario 2 due to a larger % loss in Scenario 1. For high bit-rate codestreams, the verification rate will be lower in Scenario 4 compared to Scenario 3 even though Scenario 3 has the largest β. This is due to the smaller MTU size in Scenario 3 where even a larger β does not adversely affect the % loss as that in Scenario 4. The above observation is in agreement with the results of the following computer simulations.

The verification rates of the proposed scheme in all scenarios are simulated, with the following assumptions:

1. The channel does not have a retransmission mechanism.

2. Large NALUs are divided into small packets according to the maximum transmission unit (MTU).

3. The entire NALU will be discarded when one or more of its packets are lost.

4. Unless otherwise stated, the GOF size $|\mathsf{GOF}| = 32$. Also, it is assumed that the scheme in [56] computes one signature over 5 GOFs.

5. Since the SEI NALU containing S carries the temporal layer hashes and the GOF signature, it is transmitted multiple times to increase the probability of it being received. As shown in Figure 4.6, repeating this NALU three times guarantees at least a 0.98 probability of it being received in all four scenarios.

6. Since DECS is used in each temporal layer to increase the probability of a temporal layer being authenticated, the DECS parameter k_l for each temporal layer TL_l that maximizes this probability is chosen. Figure 4.7 shows the probabilities of authenticating temporal layers in Scenario 1, which has the worst performance among the four scenarios. From the figure, setting $(k_1, k_2, k_3, k_4, k_5, k_6) = (1, 1, 1, 2, 5, 10)$ gives a ≈ 0.98 probability of authentication, where $k_1 = k_2 = 1$ is due to the hierarchical prediction structure in H.264/SVC.

In the following, the loss-resiliency, communication overhead, computation cost and buffer size requirement of the proposed scheme are examined, assuming the parameters calibrated above.

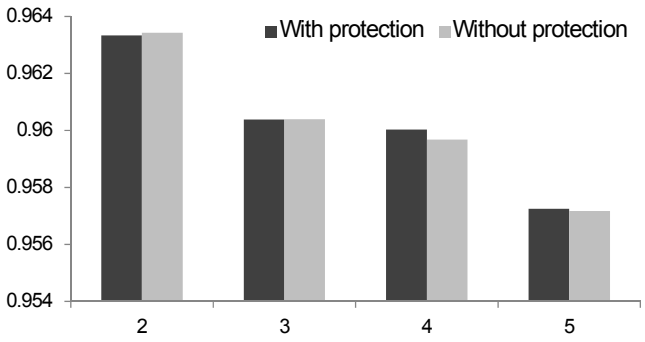


Figure 4.8: Verification rate with and without SQ NALU hash protection vs. number of SQ layers.

4.3.2 Loss resiliency w.r.t. number of SQ layers

To verify the authenticity of a frame, the frame hash must be reconstructed from hash-chaining the frame's SQ NALUs. However, in the event of loss, the hash chain can be broken. An intuitive solution is to protect the SQ NALU hashes using DECS . More specifically, for each $\mathsf{EL}_{i,j}$ (i.e. the j^{th} SQ EL NALU of F_i), use DECS to protect $h_{i,j+1} \| h_{i,j}$ such that if $\mathsf{EL}_{i,j}$ is lost, $h_{i,j+1}$ can be used to authenticate $\mathsf{EL}_{i,j+1}$ and $h_{i,j}$ can be used to reconstruct the chain. Assuming DECS computation is negligible, this method incurs a higher communication overhead, i.e. instead of having one hash per SQ NALU, there are effectively three hashes per SQ NALU.

Nevertheless, the verification rate of the proposed scheme is simulated *with* and *without* SQ NALU hash protection for Scenario 1 with the above measure, assuming that the size of every SQ layer in a frame is the same. Figure 4.8 shows that the verification rate drops when the number of SQ layers increases. Interestingly, the verification rate is not adversely affected whether or not the SQ NALU hashes are protected. Thus, for the interest of communication overhead, SQ NALU hashes are left unprotected. However, note that in [56], every SQ NALU hash is replicated twice to counter packet loss.

Next, the verification rate of the proposed scheme is compared with the scheme in [56] for different SQ layer sizes within the frames of a codestream. Assuming that there are two spatial layers, where the size of the BL is $|\mathsf{F}|/2^x$ for $x = 1, \cdots, 6$. To illustrate this, Figure 4.9 shows the relative size of the spatial BL compared to

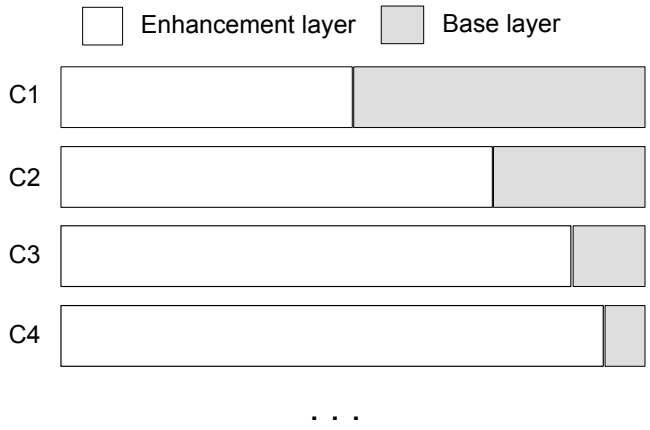


Figure 4.9: Different SQ layer size for cases C1, C2, C3 and C4.

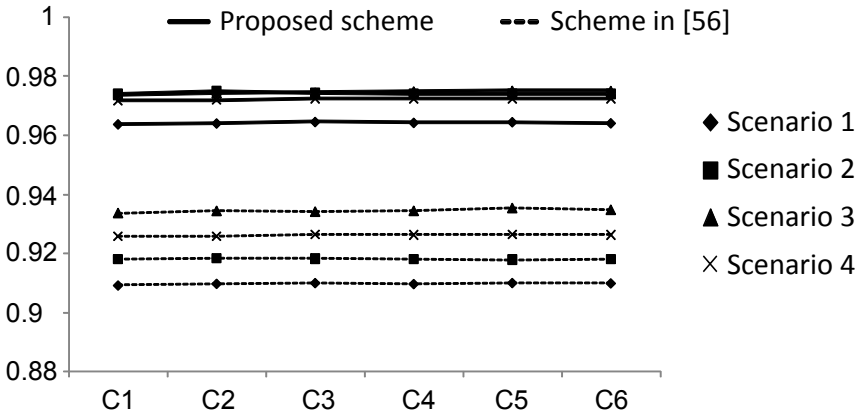


Figure 4.10: Verification rate for different SQ layer sizes defined in Figure 4.9.

the spatial EL for $x = 1, \cdots, 4$. Let Ci denote the case for $x = i$. In Figure 4.10, it is shown that the verification rates for both schemes do not vary significantly under different SQ layer sizes in all scenarios. One possible reason is that each SQ NALU is divided into several segments according to the MTU size, but the resulting number of segments is far less than the average burst length. Thus, varying the sizes of SQ layers will not significantly affect the verification rate.

4.3.3 Loss resiliency w.r.t. different SQ layer structures

As mentioned in subsection 4.2, the SQ dependency structure may not be a chain but a directed acyclic graph (DAG). The consequence of a DAG structure is that an SEI NALU for a lower SQ NALU may carry more than one hashes instead of only a single 20-byte hash (computed from its immediate higher SQ NALU). This may

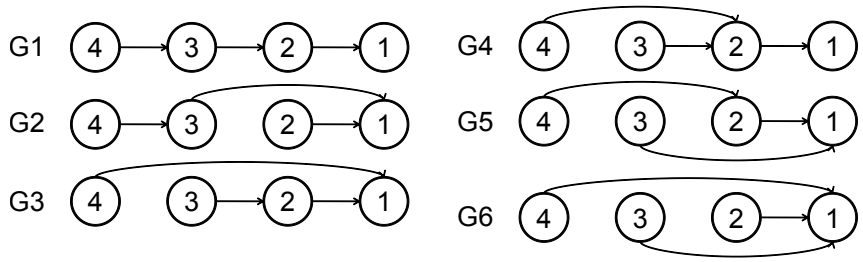


Figure 4.11: Different SQ dependency structures to be tested. Direction of arrow from A to B indicates that A is verifiable if and only if B is received and verifiable.

result in a higher loss probability during transmission, rendering the "depending" higher SQ layer unverifiable. To examine the effect of different DAG structures on the verification rate of the proposed scheme, several SQ dependency structures are defined (see Figure 4.11), and a structure i is denoted as Gi. An SQ NALU is considered verifiable if either one of the following two conditions is met:

- (Case 1) It is the SQ BL NALU - if it is received, it is verifiable
- (Case 2) It is not the SQ BL NALU - if it is received, and if the SQ NALU where its hash value is appended to is received and verifiable

Based on the above criteria, it can be seen that case G1 is the most vulnerable to packet loss because if any one of the SQ NALUs is lost, then the entire chain is unverifiable. On the other hand, case G6 is the most resilient to packet loss - as long as the SQ BL NALU is received, then any one of the SQ EL NALU is verifiable as long as it is received. Figure 4.12 shows the verification rate of the proposed scheme under different SQ dependency structures, namely from case G1 to G6. Interestingly, the verification rate does not vary significantly for different DAG structures in each scenario. One possible explanation is that since there are only four SQ layers, the maximum number of hashes an SEI NALU will carry is three. As the MTU size is set to 300 bytes (minimum), each SEI NALU can carry up to 15 hashes without being segmented, thus, as long as the SEI NALU is received, the SQ NALUs are verifiable.

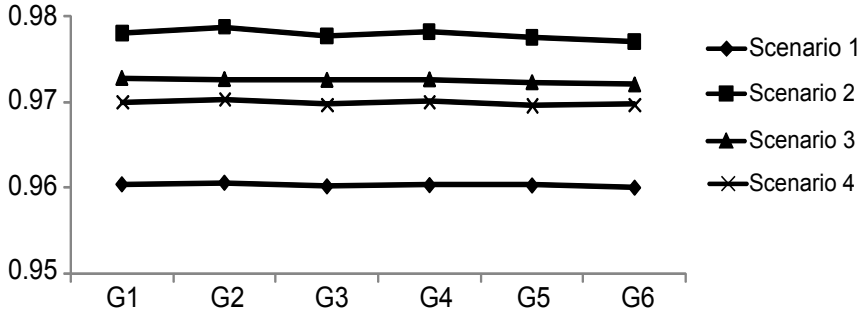


Figure 4.12: Verification rate of the proposed scheme under different SQ dependency structures defined in Figure 4.11.

4.3.4 Loss resiliency w.r.t. number of temporal layers

In the following, the verification rates of the proposed scheme and the scheme in [56] for different number of temporal layers (i.e. different GOF sizes) are examined. Since the best coding efficiency is achieved for number of temporal layers between four to six layers, i.e. GOF sizes between 8 and 32 [75][82], these cases and their impact on the verification rate of the proposed scheme and the scheme in [56] are simulated.

As shown in Figure 4.13, the verification rate of the proposed scheme in all scenario is higher than that of the scheme in [56]. For both schemes, the verification rate decreases as the number of temporal layer increases. This is because as the number of temporal layer increases, the GOF size also increases while the number of NALU S that carries the signature verifying the entire GOF remains fixed at three. As such, with an increasing GOF size, the probability of receiving the NALU S decreases. However, such decrease is to a small degree, as shown in Figure 4.13.

4.3.5 Loss resiliency w.r.t. transmission order

The verification rate of the proposed scheme by considering the codestream transmission order is further studied. For an H.264/SVC codestream generated by JSVM (Joint Scalable Video Model) [69], there are two types of codestream orders: 1) video playback (Tx_Play) and 2) preorder traversal (Tx_PT) (see Figure 4.14 and

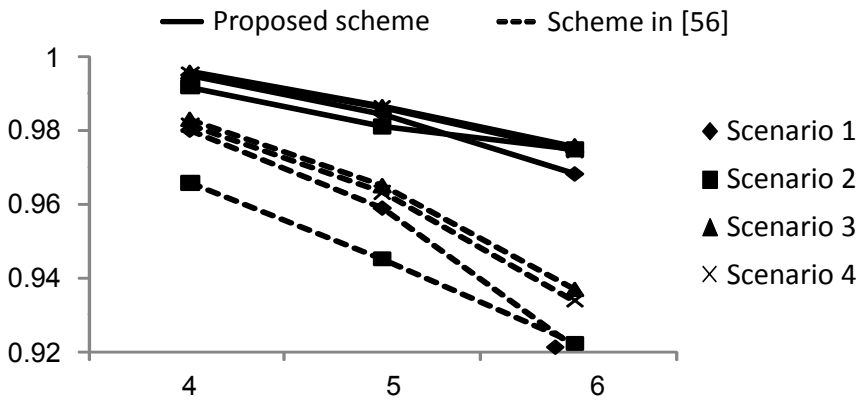


Figure 4.13: Verification rate of the proposed scheme and the scheme in [56] under different number of temporal layers.

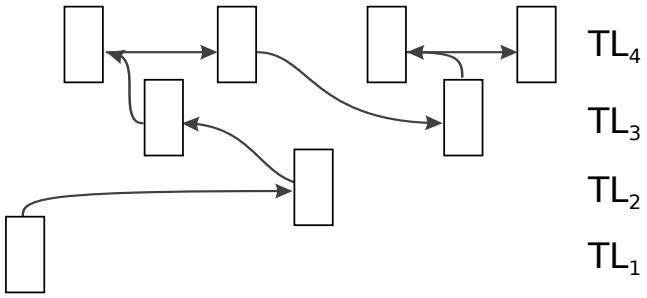


Figure 4.14: Preorder traversal, Tx_PT

4.15) based on source's encoding delay configuration D_{enc}. When D_{enc} is small, frames will be encoded and transmitted in Tx_Play order; when D_{enc} is large enough, Tx_PT order will be used. Normally, Tx_PT is used because it has less intra-frame redundancy.

For the sake of comparison, the verification rates of the proposed scheme and that of the scheme in [56] in both transmission orders are studied. Note that unlike [56], the proposed scheme does not support MGS scalability. Therefore, in the

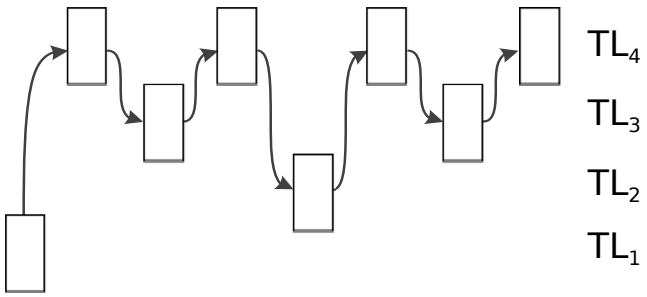


Figure 4.15: Video playback order, Tx_Play

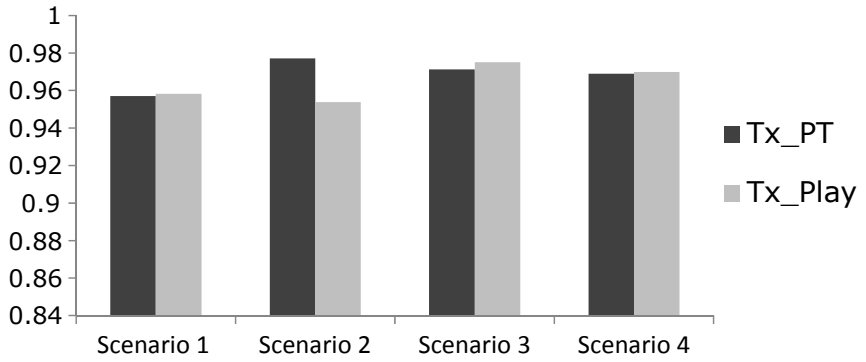


Figure 4.16: Verification rate of the proposed scheme under different transmission orders.

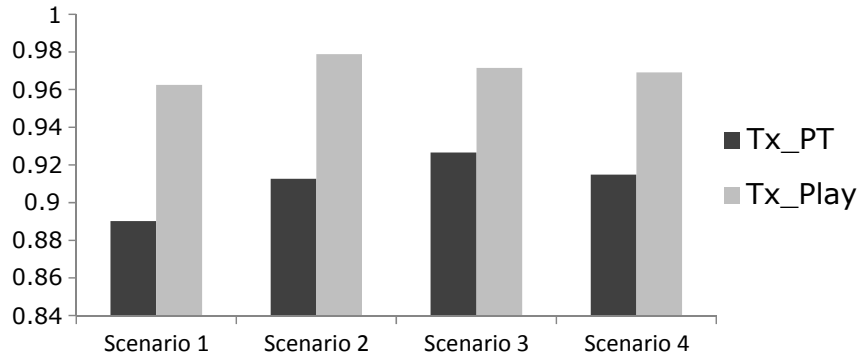


Figure 4.17: Verification rate of the scheme in [56] under different transmission orders.

implementation of the scheme in [56] (and also [58]), MGS scalability is not considered. Figures 4.16 and 4.17 show that with five SQ layers, both schemes have similar verification rates for Tx_Play while the proposed scheme has a higher verification rate for Tx_PT. This is because as mentioned earlier, the scheme in [56] authenticates temporal layers using a hash chain where the final hash is signed. Note that in [56], the "temporal base layer" is a combination of TL_2 and TL_1. In Tx_PT, the frames in TL_2 and TL_1 are transmitted consecutively, which means that they are likely to be lost in one burst. Moreover, since temporal layers are authenticated using a hash chain, the loss of the lower layers renders the higher layers unauthenticated.

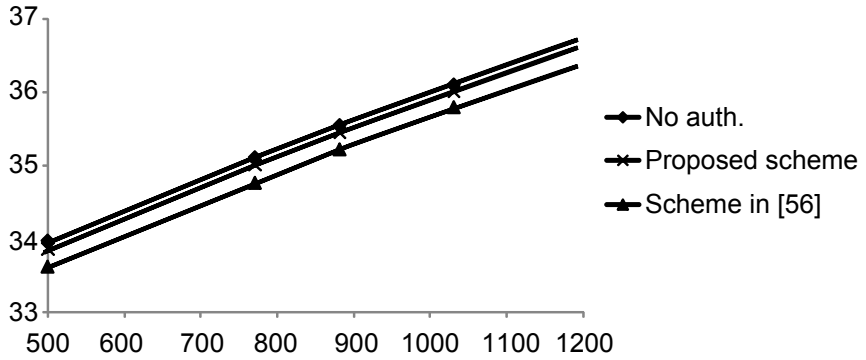


Figure 4.18: PSNR for the proposed scheme and the scheme in [56] vs. bit-rate.

4.3.6 Communication overhead

The proposed scheme achieves a high loss-resiliency with a sufficiently low communication overhead. Considering a GOF size of 32 and that each frame has five SQ layers (including SQ base layer), the proposed scheme appends a total of 3808 bytes per GOF while the scheme in [56] appends a total of 11620 bytes per GOF; for a codestream with a specific bit-rate, the amount of authentication data results in a lower effective video bit-rate, thus a lower PSNR (see Figure 4.18).

4.3.7 Computation Cost and Buffer Size

The primary computation cost is incurred by the verification delays for the digital signature algorithm and hash function. The RSA signature verification delay on a Samsung S5830@800MHz smart phone is 5.65ms per verification while SHA-1 computation time is proportional to the codestream bit-rate since hash function processes a block of 64-bit input at a time. Therefore, for $|\mathsf{GOF}| = 32$, the proposed scheme's verification delay is about 5.92ms to 6.72ms per GOF on the smart phone when the bit-rate is varied from 500 kbps to 2000 kbps. When the codestream "Foreman" is encoded with one quality enhancement layer at resolution 352×288 on this platform, the decoding time is about 85ms per frame. The proposed scheme only costs $\frac{6.72}{85 \times 32} = 0.2\%$ of the decoding time. Though the scheme in [56] incurs even shorter verification delay because there is only one signature verification per five GOFs, the verification delays in both schemes are negligible in a typical smart

	Source buffer size	User buffer size	User verification cost w.r.t. decoding time
Proposed scheme	1 GOF	1 GOF	0.2%
Scheme in [56]	5 GOFs	$1-5$GOFs	0.08%

Table 4.4: Comparison of source and user delays and computation cost.

device today. Table 4.4 compares the required buffer sizes at the source and user,and the user verification cost (compared to decoding time) of the proposed scheme with those of the scheme in [56].

4.4 Discussions

A cryptographic-based authentication scheme for H.264/SVC video codestreams over packet-lossy networks is proposed. The scheme preserves the scalability structure of H.264/SVC codestreams in the spatial, quality and temporal dimensions and have several highly desirable features such as proxy-transparency, low communication overhead and low source and user computation costs and buffer requirements. Packet loss resiliency of the scheme was studied by simulating transmission of authenticated codestreams over a realistic bursty wireless mobile network characterized using a Gilbert model. Experimental results showed that the proposed scheme achieves high verification rate over typical non-stationary, bursty packet-lossy wireless mobile networks. The proposed scheme was implemented and integrated with an SVC decoder on an Android platform, and the measurement indicated that the computation cost due to authentication is negligible.

Chapter 5

Content-based Authentication for Non-Scalable Codestreams

5.1 Introduction

Content-based authentication schemes authenticate the semantic meaning of a multimedia object by extracting an invariant *feature* from the object and computing authentication data (using keyed-hash function or digital signature algorithm) on the feature. The integrity of the object can be verified as long as its feature (i.e., semantic meaning) is unchanged. As transcoding is a *content-preserving manipulation*, content-based solutions are more efficient in authenticating transcoded non-scalable codestreams compared to cryptographic-based solutions.

It is worth noting that earlier work on content-based authentication first focused on the authentication of images. In [8] and [36], the proposed scheme extracts and authenticates features from the transform coefficients of JPEG and JPEG 2000 images, respectively. These schemes have been proven to be efficient and are able to detect semantic-changing attacks on authenticated images. Since video is a sequence of frames, and each frame is essentially a still image, many existing content-based authentication schemes for video codestreams adopt a similar design convention as that for images. The work of [18][44][81], for example, extract a feature

from the frame's coefficients (hereinafter called the *payload*), and show that the feature can be used to detect semantic-changing attacks while remain unchanged under content-preserving manipulations such as transcoding. For applications such as surveillance videos that may lose vital details if transcoded, the work of [35][63] extract a fragile feature from the frame *header* and show that both semantic-changing attacks and transcoding cause an avalanche change on the header parameters that inevitably destroys the feature.

Existing content-based authentication schemes for non-scalable codestreams may perform feature extraction in either the pixel, transform or codestream domain, as shown in Figure 5.1. A pixel-domain content-based authentication scheme takes an input frame and extracts a pixel-domain feature from the frame. The feature extracted may be structural information such as edges [17], means of pixels in 4×4 downsampled blocks [80] or optical flow of motion between frames [70]. The main advantage of pixel-domain content-based authentication schemes is the robustness of the feature against transcoding, as important structural information are commonly preserved after transcoding. However, they are more computationally intensive since a user needs to fully decode the codestream before performing verification. In addition, feature extraction in the pixel domain involves executing a feature detector algorithm (e.g., Canny [9][17] or SIFT [50][70]), and processing the detected feature using methods such as designing a Vector Quantization codebook [80] or histograms of orientation of optical flow [70] in order to reduce the feature size.

In retrospect, codestream-domain content-based authentication schemes such as those in [55] [107] [108] work directly at the codestream level. This is achieved by taking as input a watermark and the entropy-encoded codestream and identifying the set of codewords used in the codestream. Then, by designing a mapping between the used and unused codewords in the valid codespace, the used codeword is either replaced by the unused codewords or kept unchanged depending on the watermark bit. The main advantage of codestream-domain content-based authen-

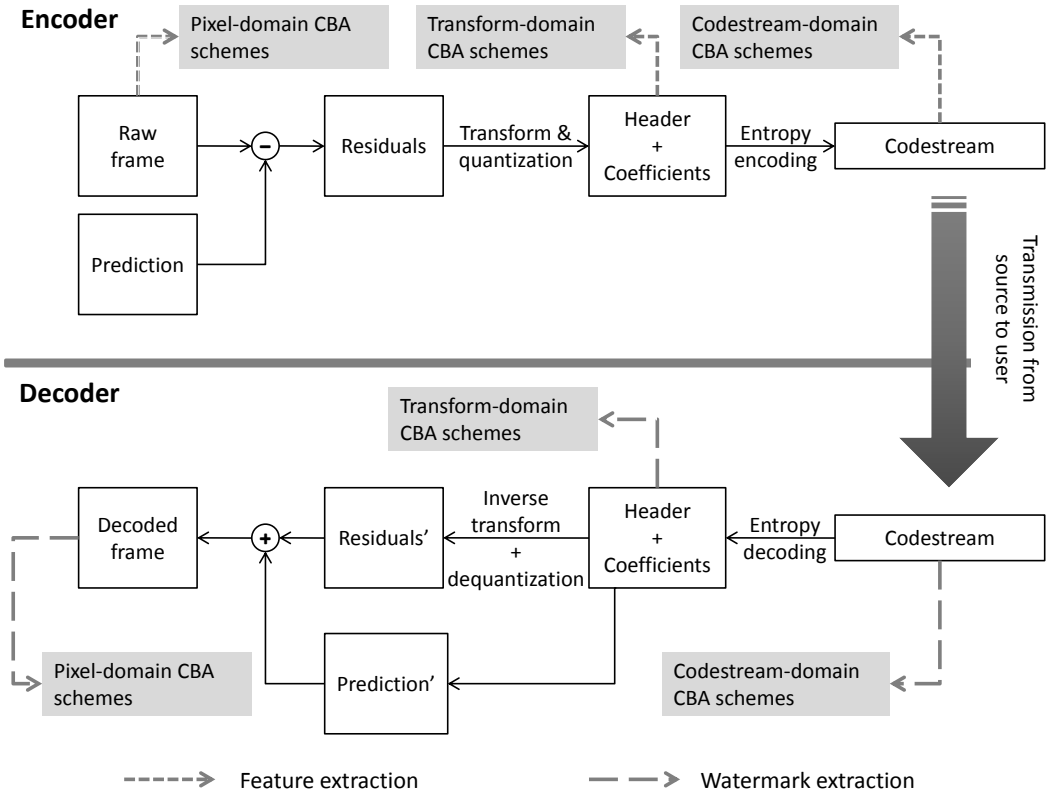


Figure 5.1: Three types of content-based authentication schemes.

tication schemes is the efficiency since verification can be performed directly at the codestream level. However, the schemes are very sensitive to random errors and transcoding; in addition, they require transmission of a detection metadata that specifies the locations to allow correct watermark extraction. Care must also be taken in order for the schemes to remain format compliant with minimal distortion after codewords replacement.

Transform-domain content-based authentication schemes are designed to trade off between efficiency, error-tolerance, perceptual distortion and robustness to transcoding of the pixel- and codestream-based content-based authentication schemes. In a transform-domain content-based authentication scheme such as those in [18][44][81], the source extracts an invariant transform-domain feature from the transform coefficients or header parameters and computes the authentication data, which is then embedded back to the transform coefficients as a watermark. During verification, a user extracts feature from the received codestream, and veri-

fies it against the extracted authentication data (i.e., watermark). Since transform-domain parameters may change as a result of coding inconsistencies introduced during transcoding, the transform-domain feature is not as robust to transcoding as a pixel-domain feature. However, transform-domain content-based authentication schemes are more efficient compared to pixel-domain schemes since the user performs only partial decoding in order to perform verification. Unlike codestream-domain content-based authentication schemes, the feature extracted in a transform-domain scheme is less sensitive to transmission errors while the watermark embedding in the transform coefficients provides robustness to both perceptual distortion and transcoding.

In this chapter, the existing transform-domain content-based authentication schemes proposed for non-scalable codestreams are reviewed and a common security flaw in these schemes that can be exploited to mount semantic-changing attacks in the transform domain is pointed out. Unlike images, where the payload (coefficients) represent an image's overall semantic meaning, the transform-domain payload and the header of a video have a strong interdependency relationship. This relationship, when maliciously exploited, changes the semantic meaning of the final, decoded multimedia representation to a similar effect as attacks in the pixel domain, and these attacks cannot be detected by existing schemes. The ways to exploit such relationship are discussed and several attack examples as the results of manipulating transform domain parameters are presented. Finally, an in-depth discussion on the attacks that manipulate the header parameters, and the condition of the attacks, given the attacker's desired attack content, is presented. Note that although the attacks are performed on H.264/AVC-encoded codestreams, they are also applicable to videos encoded by other standards such as MPEG-2 and MPEG-4 due to the same underlying video coding concept.

5.1.1 Transform-Domain Syntax of an H.264 Macroblock

In the H.264 standard, a prediction model takes as input a raw video frame and outputs a residual frame. The raw frame is first partitioned into units (each of size 16×16 pixels) called macroblocks, which may be further partitioned into 16 (4×4)-blocks. Given a raw macroblock Ori, the prediction model searches for the most perceptually similar macroblock within a *searchable region*, i.e., neighbouring macroblocks in the same frame (intra prediction) or in adjacent frames (inter-prediction), and uses the most similar macroblock as reference to generate a prediction macroblock Pred. The prediction macroblock is (pixel-wise) subtracted from the raw macroblock to obtain the residual macroblock Res as in Equation 5.1. The residual macroblock is then transformed, quantized and entropy encoded to the codestream domain.

Figure 5.2 shows the transform-domain syntax of an H.264 macroblock. In this figure, parameter *type* indicates whether the macroblock is intra- or inter-predicted. Each (intra/inter) macroblock can be partitioned into sub-blocks of different sizes, which is conveyed by the parameter *partition size*. For an intra macroblock, *prediction mode* conveys the Directional Prediction Mode (DPM) indicating the location of reference macroblock(s) and the method of generating prediction macroblock; for an inter macroblock, this parameter conveys the reference frame index pointing to a previously-decoded frame and the Motion Vector (MV) indicating the displacement of the reference macroblocks from the raw macroblock. *Coded Block Pattern* (CBP) indicates the existence of non-zero coefficients in the macroblock, followed by the *Quantization parameter* (QP). In the remainder of this chapter, these prediction parameters are collectively referred to as the macroblock *header* whereas the quantized luma and chroma coefficients are referred to as the macroblock *payload*.

At the decoder, the decoded macroblock Dec is obtained as in Equation 5.2, after reconstructing the prediction macroblock Pred^* (using the macroblock header) and the residual macroblock Res^* (using the macroblock payload). Note that for a

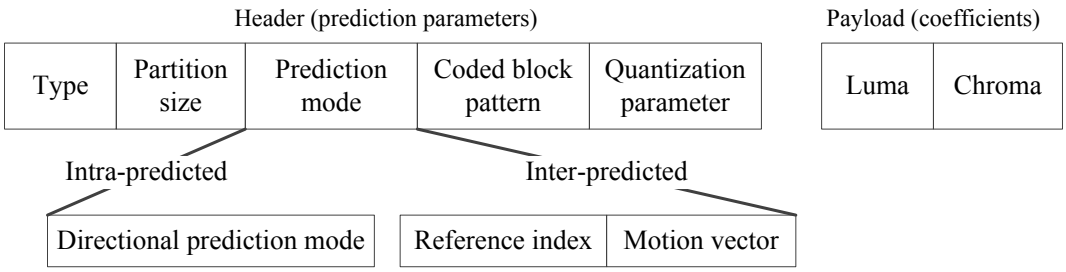


Figure 5.2: The transform-domain syntax of an H.264 macroblock.

non-tampered macroblock, the quantity α is due to lossy compression and is negligible, and Dec is perceptually similar to Ori. From Equation 5.2, an interdependent relationship between the macroblock header and payload can be observed; the ways to exploit this relationship to achieve semantic-changing attacks in the transform domain are discussed and demonstrated in the following sections.

$$\text{Encoder: } \mathsf{Res} = \mathsf{Ori} - \mathsf{Pred} \tag{5.1}$$

$$\text{Decoder: } \mathsf{Dec} = \mathsf{Pred}^* + \mathsf{Res}^* = \mathsf{Ori} + \alpha \tag{5.2}$$

5.1.2 Content-Based Authentication Model

In this subsection, a generic content-based authentication (CBA) model which is followed by most of the CBA schemes in the literature is described. A transform-domain CBA scheme for video works at the *macroblock* level, in compatible with video coding standards such as MPEG-2, MPEG-4 and H.264 that use block-based coding. Given a macroblock in the transform domain, a CBA scheme first identifies the feature extraction domain and the prediction parameter(s) or coefficients to extract feature F from. The feature F, together with the source's private key sk, serve as inputs to the feature authentication phase that outputs a watermark W_F. In the watermark embedding phase, a different secret key k is used to identify a set of embedding locations and W_F is embedded into the macroblock following a set of embedding rules. The watermarked macroblock is then entropy encoded

into a codestream and transmitted to a user. Upon receiving the codestream, the user performs entropy-decoding and watermark extraction by identifying the extraction domain, locations and extraction rules to output the watermark W_F. The user then performs the same feature extraction operation to output a feature F' of the macroblock and verifies it against W_F using the source's public key pk (which corresponds to the source's private key sk) in the feature verification phase. Upon successful verification, the user proceeds to decode the macroblock.

5.2 Classification of Existing Schemes

Existing CBA schemes for non-scalable codestreams are classified into two categories, namely payload- and header-protected CBA schemes.

5.2.1 Payload-Protected Schemes

Payload-protected schemes extract and authenticate a feature from the macroblock payload (i.e., coefficients) that can be either robust or fragile to transcoding (as surveyed in Chapter 2). The watermark computed from the feature is embedded back into the coefficients in the payload, or into the prediction parameters in the header. For embedding into payload, the rule of evaluating LSB [18][81][88][100], zero/non-zero coefficients [102] or energy relationship between coefficients [14][88] are used, whereas for embedding into header, the rule of evaluating LSB [44][74] of MVs is used.

5.2.2 Header-Protected Schemes

In the work of [35][63], a feature is extracted, respectively, from the DPMs of intra frames and the partition sizes of macroblocks. Their schemes are shown to reliably detect semantic-changing attacks as well as unauthorized transcoding due to the fragile nature of header parameters. The watermark is embedded into the payload

using the LSB evaluation rule due to limited embedding capacity in the header.

Remarks. Note that there are several CBA schemes that extract feature from the payload *and* the motion vectors [45][76][100] in the header. For clarity sake, they are not classified but as will be discussed and shown in the remainder of this chapter, almost all prediction parameters in the header have interdependent relationship with the payload that can be exploited to achieve semantic-changing attacks; these schemes are still susceptible to attacks in the transform domain.

5.3 The Design Flaw and its Exploitation

The common flaw of existing transform-domain content-based authentication (CBA) schemes is that the feature extracted is insufficient to truly represent the video semantic. This is because they did not take into account the interdependent relationship between prediction parameters in the macroblock header with the coefficients in the macroblock payload. By exploiting this relationship, attacks performed in the transform domain can not only change the codestream semantic, they are also undetectable by the CBA schemes.

5.3.1 Exploiting the Flaw in Payload-Protected Schemes

Unlike images where image pixels were directly transformed and quantized [77], a video's macroblock coefficients convey the *relationship* between the macroblock pixel content and its prediction macroblock, i.e., the residual macroblock Res. If an attacker finds an *attack prediction macroblock* Pred' to replace the *original prediction macroblock* Pred^*, the targeted macroblock Dec could be modified to the attacker's desired attack macroblock Dec' (see Equation 5.2). The discussion hereafter is based at the (4×4)-block level since it is the smallest coding unit.

To find an attack prediction block, an attacker proceeds as follow. Firstly, identify the "searchable region" and the candidate reference blocks that generate the

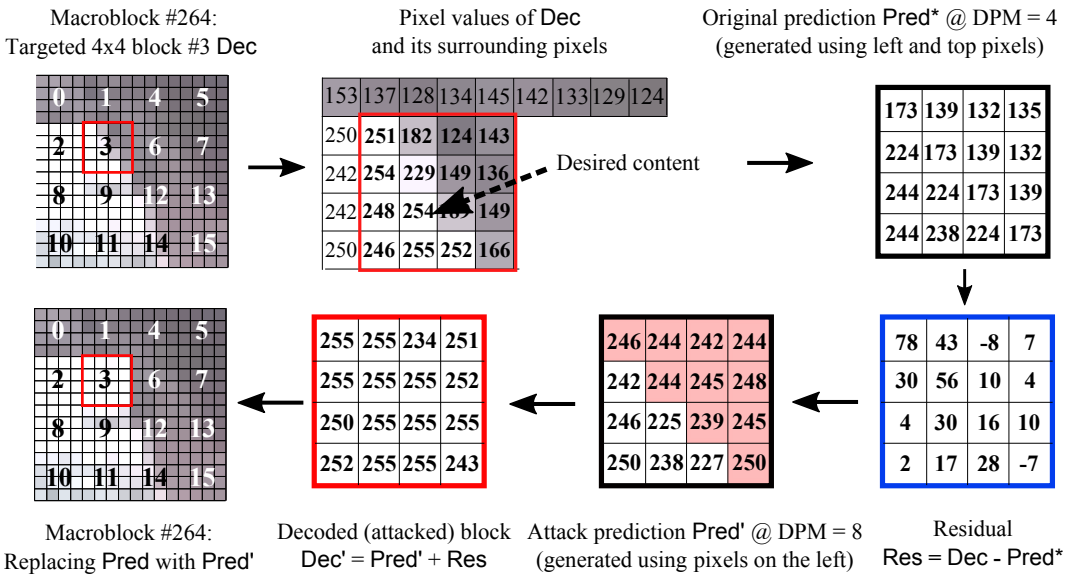


Figure 5.3: Example of finding an attack prediction block Pred′ by modifying DPM value of a targeted block Dec in order to generate the desired attack block Dec′; macroblock #264 is extracted from the Bridge sequence for content removal.

suitable Pred′ to obtain Dec′. In intra-prediction, the searchable region is the four neighbouring blocks (left, above-and-to-the-left, above, and above-and-to-the-right of) the targeted block whereas in inter-prediction, the searchable region is within an area centering on the targeted block [104]. To replace Pred* with Pred′, modify the *prediction mode* (e.g., DPM, MV and reference frame index) of the targeted block Dec. Figure 5.3 shows an example of finding an attack prediction block by modifying the DPM of a targeted intra-predicted block in the "Bridge" sequence (refer to Figure 5.7).

Depending on the video content, it is possible that a suitable Pred′ is unavailable. If so, a workaround that indirectly modifies the residual block Res without being detected by payload-protected schemes can be performed using the effect of QP. At the encoder, a larger QP in forward quantization removes insignificant coefficients. At the decoder, given a set of coefficients, a larger QP in inverse quantization magnifies the residual samples whereas a smaller QP suppresses the samples. If a decoder receives a corrupted QP, inverse quantization results in a different set of coefficients that may misrepresent the residual samples in Res. Note that this cannot be detected by payload-protected schemes because the magnifying/suppressing happens during

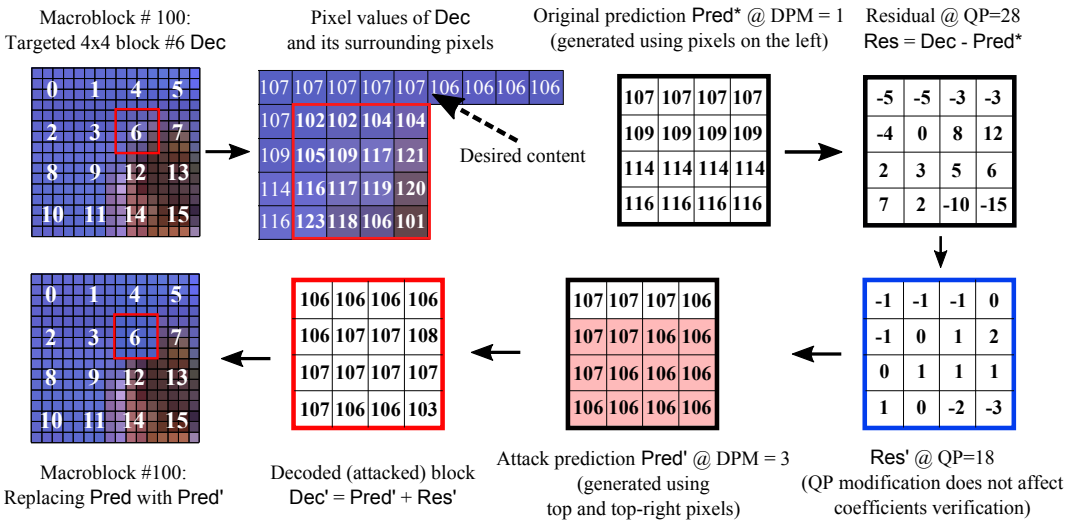


Figure 5.4: An example of finding an attack prediction block Pred′ by modifying DPM and QP values of a targeted block Dec in order to generate the desired attack block Dec′; macroblock #100 is extracted from the News sequence for content removal.

the decoding process, which is only executed *after* integrity verification. Having different QPs across macroblocks in a frame is not uncommon; macroblock-layer rate control in H.264 has been proven to improve coding efficiency [52] whereas earlier standards (e.g., MPEG-4) and the H.264 High Profile allow different QPs for DC and AC coefficients [19][71]. An example of such attack is illustrated in Figure 5.4 on an intra-predicted macroblock extracted from the "News" sequence (refer to Figure 5.5).

If the targeted macroblock spans across targeted and non-targeted content, it is more complicated to modify its prediction mode because the attacker needs to find a suitable attack prediction macroblock of the same size that changes only the targeted content while keeping the non-targeted content intact. By modifying the macroblock *partition size*, the targeted macroblock can be partitioned into sub-blocks, such that the targeted content is isolated in a sub-block, and then perform a search for the suitable attack prediction sub-block thereof.

Remarks. Attacks on payload-protected schemes involve replacing the original prediction block with an attack prediction block in order to change the content of a

targeted block. Given the searchable region which is constrained in one frame (intra frames) or within the same video (inter frames), arbitrary content insertion attacks cannot be realized. However, content removal and modification attacks are possible as will be shown in Section 5.4. Also note that prediction mode parameters such as DPM, MV and reference frame index are coded differentially between successive blocks. If these parameters are changed, it may affect the corresponding parameter of subsequent (targeted/non-targeted) blocks, causing them to use a wrong/different prediction block for decoding. This may result in error propagation that occur in the form of visual distortion on the decoded frame. In Section 5.4, an example of such error propagation is illustrated, and the way to correct the visual distortion by either restoring the prediction mode of affected blocks or by restricting their choice of prediction block to a more suitable one is shown.

5.3.2 Exploiting the Flaw in Header-protected Schemes

Although header-protected schemes can detect both content-preserving and semantic-changing manipulations, they are more insecure compared to payload-protected schemes. Since the payload represents the residual block with samples that are integers ranging from -255 to +255, an attacker can perform a simple but powerful attack using reverse engineering. Since the user has no prior information about the original block, an attacker can replace them with a new block with different content Dec' and compute the new residual block Res' such that $\mathsf{Res}' = \mathsf{Dec}' - \mathsf{Pred}$, where Pred is the original prediction block. The attacker then performs forward transform and quantization to obtain a new set of transform coefficients, replacing the original coefficients in the payload.

5.3.3 Complying with Watermark Extraction

Apart from ensuring that the transform-domain attacks do not alter the authenticated feature, it is also vital to ensure that the tampered data obeys the watermark extrac-

tion rule. Watermark extraction includes: extract location identification and extraction based on extraction rules. Although random extraction locations is deemed vital for security reason [53], it is more important for copyright protection where the attack objective is to find and destroy the watermark; a successful attack in the presented approach depends more heavily on complying with the watermark extraction rules. For verification efficiency, existing CBA schemes perform extraction by evaluating either the LSB or zero/non-zero coefficients as mentioned in subsection 5.2. Such characteristics can always be engineered in the coefficients or MVs. Since DPMs can be categorized into sets generating similar prediction blocks [62], an attacker can select DPMs within the same set to satisfy the even/odd evaluation.

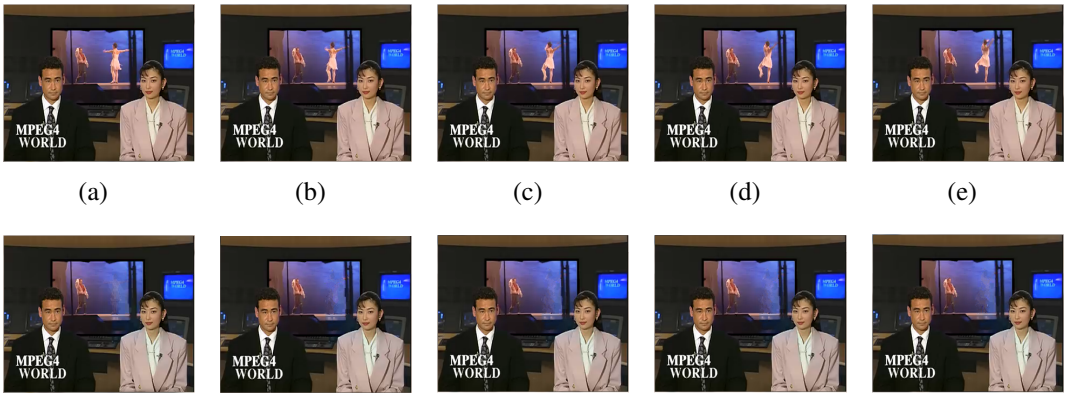
5.4 Attack Examples on Existing CBA Schemes

In this section, transform-domain attack examples that can be applied on each category of CBA schemes as discussed in Section 5.3 are demonstrated. More specifically, this section presents content removal and content modification attacks on payload-protected schemes, and content insertion attack on header-protected schemes. The attacks are implemented using the JM reference software [39], where the attacker's interception and replacement of macroblock codestream are emulated by modifying the decoder's 'read' data. The video sequences used in the attacks are the 352×288 News, Bridge and Waterfall sequences [3] and a 384×288 surveillance sequence [85], all encoded in IBBBBBBBP format with QP = 28 for intra frames and QP = 30 for inter frames. The source files can be viewed at https://sites.google.com/site/smusvc/Authentication.

5.4.1 Content Removal Attacks

A content removal attack is the act of replacing an object with its background information.

Figures 5.5(a)-5.5(e) show the first five frames of the original News sequence,



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5.5: Content removal attack on News sequence, with the original frames shown in (a)-(e) and attack frames in (f)-(j).

where Figure 5.5(a) is an intra frame and Figure 5.5(b)-5.5(e) are inter frames. The aim of the attack is to remove content of the targeted blocks, i.e., the ballerina, by finding new attack prediction blocks such that the end result is a set of attacked blocks that convey the background information, i.e., the walls. Notice that in Figure 5.5(a), the targeted blocks are surrounded by reference blocks conveying similar content, i.e., the walls. This is an example where the attack prediction blocks *are* the original prediction blocks and it implies that the samples in the (targeted) residual blocks have high magnitude since they do not have similar prediction blocks to be used for compression (see Equation 5.1). Hence, the workaround by manipulating QP of the targeted blocks to suppress their residual samples is executed. Subsequently, if necessary, the DPMs of the targeted blocks (e.g., torso of the ballerina) are modified to use background blocks as attack prediction blocks.

Since intra frames are used as (one of the) reference frame(s) to generate prediction blocks for the subsequent inter frames, the content of the attacked intra frame will "propagate" to the inter frames during decoding. The residuals of the original content in inter frames were completely removed by modifying the MV of targeted blocks in the inter frames. The final result of the removal attack is shown in Figure 5.5(f)-5.5(j).

Due to differential coding of prediction mode parameters, there is a risk of er-

(a) Distorted frame (b) Corrected frame

Figure 5.6: Visual distortion due to DPM decoding error and its correction.

ror propagation after one is manipulated. Figure 5.6(a) shows an example of error propagation due to erroneous DPM decoding in an intra frame, which is resolved by correcting the DPM of the affected block(s) to use a *more* suitable prediction block for decoding. The result of this correction is shown in Figure 5.6(b).

There are also cases where QP manipulation is not needed. Figure 5.7(a) shows the first frame of the Bridge sequence. In this example, it is sufficient to modify the DPMs of targeted blocks, i.e., the left pier, to use the background information, i.e., the river, as attack prediction blocks. The result of the removal attack is shown in Figure 5.7(b). In this case, QP manipulation is not needed because the original prediction blocks are obtained from the top of the targeted blocks and they are semantically similar, thus, the residual blocks have samples of small magnitude. Replacing the original prediction blocks with the attack prediction blocks on the left (i.e., the river) replaces the content of the targeted blocks with the content of the attack prediction blocks.

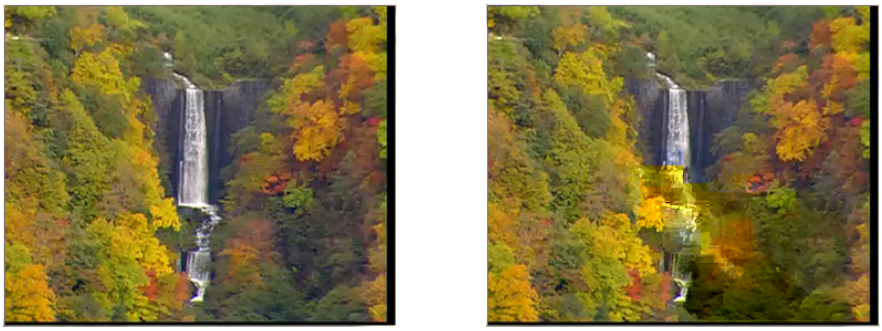
5.4.2 Content Modification Attacks

In this subsection, two examples of content modification attacks on payload-protected schemes that includes content replacement and content relocation attacks are shown.

Content replacement is the act of replacing (or "overwriting") the content of a

(a) Original frame (b) Attacked frame

Figure 5.7: Content removal attack on Bridge sequence.



(a) Original frame (b) Attacked frame

Figure 5.8: Content replacement attack on Waterfall sequence..

targeted block with that of its desired attack block. In the first example, content replacement attack is mounted on the intra frame of the Waterfall sequence. As shown in Figure 5.8, the DPMs of a large set of targeted blocks are modified to "extend" the effect of attack prediction blocks, i.e., the trees, such that they cover the original blocks, i.e., the waterfall.

In the second example, the reference frame index is modified to achieve content replacement in inter frames. In addition, the partition size parameter is also modified to facilitate the attack. Figure 5.9(b) shows the timing information extracted from a surveillance frame in Figure 5.9(a). This timing information is encoded using 16×16 macroblocks, where the upper half of each macroblock covers the timing information (targeted) while the lower half covers the surveillance background (non-targeted). Tampering with the reference frame index will affect *both* the timing information and the surveillance background. By manipulating the *partition size* pa-

(a) Original frame (b) Original timing (c) Attacked timing

Figure 5.9: Content replacement attack on a surveillance sequence.

rameter such that each targeted 16×16 macroblock is partitioned into sixteen 4×4 blocks, the targeted content is isolated from the non-targeted content. The reference frame index of the targeted blocks can then be modified independently without affecting the non-targeted blocks. Figure 5.9(c) shows the result of this attack; when the attacked frames are inserted into the video sequence, a scrambled timing information is observed.

Content relocation is the act of changing the position of an object from one location to another. This attack is typically difficult to achieve on intra frames because each intra block is predicted from its neighbouring blocks; to perform a meaningful content relocation that is affected by, and will be affecting, neighbouring blocks is intuitively hard. For an inter block, however, this attack can be achieved by modifying the MV using a concept similar to content removal. Figure 5.10(a) shows a frame extracted from the surveillance sequence. In this attack, the MVs of the targeted blocks, i.e., the dustbin, are modified such that they use a new content, i.e., the man, as attack prediction blocks. Subsequently, the blocks containing the man is removed using the content removal attack methodology presented in the previous subsection. The result of this attack is shown in Figure 5.10(b).

5.4.3 Content Insertion Attacks

For completeness, an example of content insertion attack on header-protected schemes is shown since this attack is not possible on payload-protected schemes. Figure 5.11 shows an example of content insertion attack on header-protected CBA

(a) Original frame (b) Attacked frame

Figure 5.10: Content relocation attack on a surveillance sequence.

Figure 5.11: Content insertion attack on header-protected CBA schemes.

schemes, where the original frame is shown in Figure 5.9(a). Taking the samples of arbitrary image of a clock, the residual blocks are obtained by subtracting the original prediction blocks from the samples. The residual blocks are then transformed and quantized, and inserted into the macroblock payload.

5.4.4 Summary and Remarks

In summary, in contrary to images, videos' transform-domain header parameters and payload must be simultaneously integrity protected since their interdependency relationship can be exploited to mount semantic-changing attacks in the transform domain. For attacks on payload-protected schemes, DPM, MV and reference frame index affect the generation of prediction block, which when combined with the (untampered) residual block, could semantically change the targeted block. While DPM selects prediction blocks from neighbouring blocks, MV and reference frame index select them from a wider search range. An advanced attacker may modify the

macroblock type (intra/inter) and remove or insert bogus prediction mode relevant to the new macroblock type; attacks of such nature are left as future work. Additionally, the QP is a header parameter that can be used as a workaround to inexplicitly modify the residual block while the partition size can be modified to facilitate search for a suitable prediction block. For attacks on header-protected schemes, it is vital that the distribution of tampered coefficients tallies with the coded block pattern (CBP) in the header, otherwise a decoding error may occur. It is acknowledged that authenticating the CBP in the header will impose a higher level of difficulty on the attacks, however, in existing header-protected schemes, this parameter is often left unprotected. In the literature, there are also CBA schemes that authenticate both the payload and the MVs in the header [45][76][100]. However, as shown in the attack examples, these schemes are still vulnerable to attacks such as DPM attacks on intra blocks, reference frame index and/or partition size attacks on inter blocks.

For H.264/SVC, i.e., the scalable extension of H.264/AVC that is used to encode the sequences used in this study, a mandatory base layer (BL) that is backward compatible with AVC is encoded. Using BL as reference to generate prediction, one or more enhancement layers (ELs) that gradually improve the resolution or quality of the video are encoded. If header-protected CBA schemes are applied on an SVC codestream, attacks on the payload of BL and ELs are possible (and powerful). On the other hand, if coefficients-protected CBA schemes are applied, the attacks presented in the previous section are applicable to the BL and the effect could propagate to the ELs. Thus, noting the importance of the BL, the work of [89] cryptographically protects the BL to prevent any form of malicious tampering[1]. Although there are minimal header parameters in the ELs [75], there are still several important parameters, e.g., the *motion prediction flag* and *residual prediction flag*. For the ELs, a motion prediction flag of '1' indicates that the EL directly uses header parameters of its reference (base) layer; otherwise, it carries its own header parameters. A

[1]Pixel-domain CBA scheme is used in [89] to protect the ELs and thus is out of the scope of study for this study.

residual prediction flag of '1', on the other hand, indicates that the EL's payload R'_{EL} is obtained by subtracting the upsampled BL payload R_{BL} from the payload obtained via AVC-like encoding R_{EL}; otherwise, $R'_{EL} = R_{EL}$. An advanced attacker could then opt to modify these flags and to manipulate the video semantic. In short, content-based authentication for SVC present several interesting research problems to be explored.

5.5 Analysis on the Attacks on Payload-Protected Schemes

Semantic-changing attacks on videos authenticated by payload- or header-protected CBA schemes are possible by modifying, respectively, the header or the payload of the targeted block(s); moreover, these attacks cannot be detected by the respective CBA schemes.

Since the attacks on header-protected schemes are relatively straightforward, the following discussions focus on the attacks on *payload-protected schemes*. As shown in Section 5.4, a targeted block will convey a semantically different content as compared to its original content if a *suitable* attack prediction block is found from the searchable region. In this section, the ways for an attacker to obtain the suitable attack prediction block - given the attacker's desired attack block and the unmodifiable residual block - is analyzed. The analysis is performed on a 4×4-block level, where a macroblock M is represented as follow:

$$M = \begin{array}{|c|c|c|c|} \hline M(B_0) & M(B_1) & M(B_4) & M(B_5) \\ \hline M(B_2) & M(B_3) & M(B_6) & M(B_7) \\ \hline M(B_8) & M(B_9) & M(B_{12}) & M(B_{13}) \\ \hline M(B_{10}) & M(B_{11}) & M(B_{14}) & M(B_{15}) \\ \hline \end{array}$$

where $M(B_i)$ denotes the i-th (4×4)-block of M, and can be represented by a 4×4

Table 5.1: List of notations

Notations	**Descriptions**
Dec, Res, Pred	Original decoded, residual and prediction macroblock, respectively, each containing 16 4×4-blocks
$\mathsf{Dec}(\mathsf{B})$, $\mathsf{Res}(\mathsf{B})$, $\mathsf{Pred}(\mathsf{B})$	Original decoded, residual and prediction 4×4-block, respectively
$\bar{d}(\mathsf{B})$, $\bar{p}(\mathsf{B})$	Average of the samples in $\mathsf{Dec}(\mathsf{B})$ and $\mathsf{Pred}(\mathsf{B})$, respectively
$\hat{r}(\mathsf{B})$	Median of residual samples in $\mathsf{Res}(\mathsf{B})$
$\mathsf{Dec}'(\mathsf{B})$, $\mathsf{Res}'(\mathsf{B})$, $\mathsf{Pred}'(\mathsf{B})$	An attack decoded, residual and prediction 4×4- block, respectively
$\bar{d}'(\mathsf{B})$, $\hat{r}'(\mathsf{B})$, $\bar{p}'(\mathsf{B})$	Average of the samples in $\mathsf{Dec}'(\mathsf{B})$, $\mathsf{Res}'(\mathsf{B})$, $\mathsf{Pred}'(\mathsf{B})$, respectively
$\mathsf{E}(\mathsf{Res}) = \sum_{i,j=0}^{3} \frac{r_{i,j}}{16}$	Energy of the residual samples in $\mathsf{Res}(\mathsf{B})$, where $r_{i,j}$ is the residual sample at position (i, j) in $\mathsf{Res}(\mathsf{B})$
$\oplus$, $\ominus$	Pixel-/Sample-wise addition and subtraction, respectively

matrix. Using the same convention, an original and prediction macroblock (denoted Dec and Pred respectively), are made up of $\mathsf{Dec}(\mathsf{B_i})$ and $\mathsf{Pred}(\mathsf{B_i})$ for $i = 0, \cdots, 15$. The list of notations is shown in Table 5.1.

Generally, the average value of a 4×4-block is a good approximation of the block's samples [51][57][98] due to the high correlation between samples in the block. Since the residual block may consist of positive and negative integers, the median of the residual samples is used to indicate the relationship between the original block and the original prediction block. In other words, if $\hat{r}(\mathsf{B}) > 0$, then $\mathsf{Dec}(\mathsf{B})$ is perceptually brighter than $\mathsf{Pred}(\mathsf{B})$; otherwise, $\mathsf{Dec}(\mathsf{B})$ is perceptually darker than $\mathsf{Pred}(\mathsf{B})$. In addition, let $\mathsf{E}(\mathsf{Res})$ be the energy of the residual samples in $\mathsf{Res}(\mathsf{B})$ as defined in Table 5.1.

Given an original (targeted) block $\mathsf{Dec}(\mathsf{B})$ with $\bar{d}(\mathsf{B})$ and the residual block $\mathsf{Res}(\mathsf{B})$ having a median $\hat{r}(\mathsf{B})$, the conditions on the desired attack block $\mathsf{Dec}'(\mathsf{B})$ (in terms of $\bar{d}'(\mathsf{B})$) such that the attacker can find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$, where $\mathsf{Dec}'(\mathsf{B}) = \mathsf{Pred}'(\mathsf{B}) \oplus \mathsf{Res}(\mathsf{B})$ is discussed. The following analysis can be

applied to the attacks on both intra and inter blocks.

Case 1A. When most of the residual samples are positive, i.e., $\hat{r}(\mathsf{B}) > 0$, it implies that the original (targeted) block $\mathsf{Dec}(\mathsf{B})$ is perceptually brighter than the original prediction block $\mathsf{Pred}(\mathsf{B})$, i.e., $\bar{d}(\mathsf{B}) > \bar{p}(\mathsf{B})$. If the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually brighter than $\mathsf{Dec}(\mathsf{B})$, then an attacker finds an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ if and only if $\bar{d}'(\mathsf{B}) \geq \bar{d}(\mathsf{B}) + 2\hat{r}(\mathsf{B})$.

To prove this, suppose $\bar{d}(\mathsf{B}) < \bar{d}'(\mathsf{B}) < \bar{d}(\mathsf{B}) + 2\hat{r}(\mathsf{B})$. Substituting Equation 5.2:

$$\begin{aligned} \bar{p}(\mathsf{B}) + \hat{r}(\mathsf{B}) < \bar{p}'(\mathsf{B}) + \hat{r}(\mathsf{B}) < \bar{p}(\mathsf{B}) + \hat{r}(\mathsf{B}) + 2\hat{r}(\mathsf{B}) \\ \bar{p}(\mathsf{B}) < \quad \bar{p}'(\mathsf{B}) \quad < \bar{p}(\mathsf{B}) + 2\hat{r}(\mathsf{B}) \end{aligned} \tag{5.3}$$

Referring to Equation 5.3, an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ with $\bar{p}'(\mathsf{B})$ cannot be found. Otherwise, by computing $\mathsf{Res}'(\mathsf{B}) = \mathsf{Dec}(\mathsf{B}) \ominus \mathsf{Pred}'(\mathsf{B})$, the upper and lower bound of $\hat{r}'(\mathsf{B})$ is:

$$\hat{r}'(\mathsf{B})_{UB} = \bar{d}(\mathsf{B}) - \bar{p}(\mathsf{B}) = \hat{r}(\mathsf{B}), \text{ and} \tag{5.4}$$

$$\hat{r}'(\mathsf{B})_{LB} = \bar{d}(\mathsf{B}) - (\bar{p}(\mathsf{B}) + 2\hat{r}(\mathsf{B})) = -\hat{r}(\mathsf{B}) \tag{5.5}$$

In other words, $-\hat{r}(\mathsf{B}) < \hat{r}'(\mathsf{B}) < \hat{r}(\mathsf{B})$. This implies that compared to the original prediction block $\mathsf{Pred}(\mathsf{B})$, the attack prediction block $\mathsf{Pred}'(\mathsf{B})$ generates smaller residual samples if it is used to encode $\mathsf{Dec}(\mathsf{B})$. This contradicts the video coding rule, where $\mathsf{Pred}(\mathsf{B})$ is initially chosen to encode $\mathsf{Dec}(\mathsf{B})$ because it generates the smallest Sum of Absolute Errors, $\text{SAE} = \sum_{i,j=0}^{3} |d_{i,j} - p_{i,j}|$, where $d_{i,j}$ is the sample of $\mathsf{Dec}(\mathsf{B})$ at position (i, j) and $p_{i,j}$ is the sample of $\mathsf{Pred}(\mathsf{B})$ at position (i, j), compared to all other candidate prediction blocks in the searchable region [71]. This case can be demonstrated in the attack shown in Figure 5.7(b). If $\mathsf{Pred}'(\mathsf{B})$ cannot

be found, the workaround by manipulating QP can be implemented to suppress the residual samples so that the available candidate prediction blocks can be used to obtain $\mathsf{Dec}'(\mathsf{B})$.

Case 1B. When most of the residual samples are positive, i.e., $\hat{r}(\mathsf{B}) > 0$, but the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually darker than the original block $\mathsf{Dec}(\mathsf{B})$, then the minimum value for a sample $d'_{i,j}$ in $\mathsf{Dec}'(\mathsf{B})$ must be equal to the residual sample $r_{i,j}$ in $\mathsf{Res}(\mathsf{B})$. This is because the minimum sample for $\mathsf{Dec}'(\mathsf{B})$ is when $\mathsf{Pred}'(\mathsf{B}) = 0$ (see Equation 5.2), otherwise, if $\bar{d}'(\mathsf{B}) < \hat{r}(\mathsf{B})$, then by substituting Equation 5.2, $\bar{p}'(\mathsf{B}) + \hat{r}(\mathsf{B}) < \hat{r}(\mathsf{B})$ and the samples in the attack prediction block is less than zero, which is not feasible. Thus, in approximation, $\hat{r}(\mathsf{B}) \leq \bar{d}'(\mathsf{B}) < \bar{d}(\mathsf{B})$. This condition is demonstrated in the attack shown in Figure 5.5, where the background information (the walls) is perceptually darker than the targeted blocks (the ballerina), but the residuals samples are too large for $\mathsf{Dec}'(\mathsf{B})$ to satisfy this condition. The QP can then be manipulated to suppress/magnify the residual samples as deemed necessary.

Case 2A. When most of the residual samples are negative, i.e., $\hat{r}(\mathsf{B}) < 0$, the original block $\mathsf{Dec}(\mathsf{B})$ is perceptually darker than the original prediction block $\mathsf{Pred}(\mathsf{B})$. Suppose the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually brighter than $\mathsf{Dec}(\mathsf{B})$, then a sample $d'_{i,j}$ in $\mathsf{Dec}'(\mathsf{B})$ is upper bounded by $255 - |r_{i,j}|$ in $\mathsf{Res}(\mathsf{B})$ as dictated by Equation 5.2. Thus, it can be written in approximation that $\bar{d}(\mathsf{B}) < \bar{d}'(\mathsf{B}) \leq 255 - |\hat{r}(\mathsf{B})|$. A similar analysis to Case 1B can be applied, where if $\bar{d}'(\mathsf{B}) > 255 - |\hat{r}(\mathsf{B})|$, then the attacker must find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ where $\bar{p}'(\mathsf{B}) > 255$, which is not possible. This condition can be observed in the attack shown in Figure 5.8(b).

Case 2B. When most of the residual samples are negative, i.e., $\hat{r}(\mathsf{B}) < 0$, but the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually darker than the original block $\mathsf{Dec}(\mathsf{B})$, then an attacker finds an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ if and only if

Table 5.2: Summary of Cases 1A, 1B, 2A and 2B.

	$\hat{r}(\mathsf{B}) > 0$	$\hat{r}(\mathsf{B}) < 0$
$\mathsf{Dec}'(\mathsf{B})$ is perceptually brighter than $\mathsf{Dec}(\mathsf{B})$	**Case 1A** $\bar{d}'(\mathsf{B}) - \bar{d}(\mathsf{B}) \geq 2\hat{r}(\mathsf{B})$	**Case 2A** $\bar{d}(\mathsf{B}) < \bar{d}'(\mathsf{B}) \leq 255 - \lvert\hat{r}(\mathsf{B})\rvert$
$\mathsf{Dec}'(\mathsf{B})$ is perceptually darker than $\mathsf{Dec}(\mathsf{B})$	**Case 1B** $\hat{r}(\mathsf{B}) \leq \bar{d}'(\mathsf{B}) < \bar{d}(\mathsf{B})$	**Case 2B** $\bar{d}'(\mathsf{B}) - \bar{d}(\mathsf{B}) \leq -2\lvert\hat{r}(\mathsf{B})\rvert$

$\bar{d}'(\mathsf{B}) \leq \bar{d}(\mathsf{B}) - 2|\hat{r}(\mathsf{B})|$. This condition can be obtained by a similar prove by contradiction as in Case 1A, whereas an illustration example is shown in the upper torso, especially the head of the ballerina in Figure 5.5(f).

Table 5.2 summarizes the conditions for the above cases. When the attack block $\mathsf{Dec}'(\mathsf{B})$ cannot satisfy the conditions in any of the cases above, then the attacker cannot find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ such that $\mathsf{Dec}'(\mathsf{B}) = \mathsf{Pred}'(\mathsf{B}) \oplus \mathsf{Res}(\mathsf{B})$. A workaround can then be performed by modifying the unprotected QP to suppress or magnify the residual samples depending on the available candidate attack prediction blocks.

5.6 Discussion

Existing content-based authentication (CBA) schemes designed for non-scalable video codestreams are insecure due to insufficient feature extraction. The overlooked interdependent relationship between the header and payload parameters can be exploited to perform semantic-changing attacks in the transform domain. In this chapter, several semantic-changing attack examples that are performed in the transform domain are presented and these attacks cannot be detected by the schemes. This is followed by a discussion focusing on the conditions at which an attack on payload-protected CBA schemes can succeed given a desired attack content and the unmodifiable payload, and if not, a workaround for it.

A possible countermeasure to these attacks is to use more complicated watermark extraction rules. However, unlike images, real-time extraction is vital for

video authentication [66] which makes straight-forward watermark extraction rules such as those surveyed in this chapter highly preferred. Another possible countermeasure is to extract and authenticate features from both the header and payload domains. In practical applications, transcoding requires full decoding of intra frames and partial decoding of inter frames. The transcoding of intra frames drastically changes the header and payload [42], and to the best of the author's knowledge, there is no work that addresses this problem. Transcoding inter frames changes the payload while the remaining data in the header remains unchanged. Although existing payload-domain schemes are able to extract a stable feature from the coefficients, but sparsely-distributed coefficients in inter frames (especially after transcoding) are commonly overlooked, thereby leaving them vulnerable to tampering. Thus far, a stable feature from the header of intra frames is observed. The future research is to design a secure and efficient authentication scheme that overcomes the vulnerability of existing schemes and is robust against bit-rate transcoding (performed by semi-trusted intermediary proxies) as described above.

Chapter 6

Access Control for Scalable Multimedia Codestreams

6.1 Introduction

Cryptographic-based access control manages authorization to protected data by encrypting them such that only authorized users with the right access keys can decrypt the data. Access control is an important problem in multimedia applications such as IPTV and video-on-demand, where a multimedia source publishes different privilege levels, each granting its subscribers the access right to codestreams of different temporal, spatial and/or quality scalability. In multimedia coding standards such as H.264/SVC, a video is encoded into a multi-layered codestream, where decoding more enhancement layers in a specific scalability (along with the mandatory base layer) produces a multimedia presentation of higher quality in that dimension. Hence, scalable codestreams such as the H.264/SVC provides a readily available structure for a source to provide "subscription packages" with different privilege levels, where the higher privilege level a user subscribes to, the more enhancement layers the user is entitled to have access to.

An access control scheme designed for H.264/SVC codestreams should employ an access keys generation hierarchy that is compatible with the codestream

structure. The access keys generation hierarchy should also be efficient such that the source and users of different privilege levels need to maintain/obtain minimum number of access keys in order to decrypt the codestreams. In addition, the access keys generation process should be secure against collusion attack, where two or more users having access keys to codestreams of lower privilege levels collude to obtain access key to codestreams of a higher privilege level. In an access control framework, online key distribution center (KDC) is commonly employed to distribute access keys to authorized users. This poses a scalability problem when the number of users increases and improper management of online KDC could also result in a single point of failure, thus, it is desirable to eliminate the need for an online KDC.

There are only a handful of access control schemes for scalable multimedia codestreams. The work of [106] designed an access control scheme for MPEG-4 FGS (Fine Grain Scalability) codestreams. It uses independent access key to encrypt different enhancement layers. As a result, each user needs to maintain one access key for each layer in the privilege level that he/she subscribes to. To overcome this limitation, [105] proposes to use the Diffie-Hellman technique to generate access keys for all lower layers so that a user needs to only maintain a single access key regardless of privilege level he/she subscribes to. However, the scheme is vulnerable to collusion attacks as will be shown in Section 6.3. The work of [93] proposes an access control scheme for H.264/SVC codestreams, where a user needs to only maintain a single access key, but the scheme is also vulnerable to collusion attack. In addition, these schemes assume the use of an online KDC for access keys distribution.

In this chapter, an access control scheme for generic scalable codestream and H.264/SVC scalable multimedia codestreams, respectively, is proposed. The schemes use efficient access keys generation hierarchies that are secure against collusion attack, and require the user to maintain only a single access key regardless of the privilege level that a user subscribes to. The schemes eliminate the use of

an online KDC by using ciphertext-policy attribute-based encryption (CP-ABE) to encrypt the access keys.

6.1.1 Ciphertext-Policy Attribute-Based Encryption

The building block used in this chapter is the Ciphertext-Policy Attribute-Based Encryption scheme (CP-ABE) proposed in [7]. In CP-ABE, every user's secret key is associated with a set of attributes while every ciphertext is associated with a ciphertext policy, i.e. an access structure on attributes. A user successfully deciphers a ciphertext on the condition that the user's key attributes satisfy the access structure specified in the ciphertext.

Specifically, an access structure is represented by an access tree $\mathcal{T}_R$ with a root denoted by R and n leaf nodes corresponding to n attributes respectively. Each inner node x with c_x children nodes is attached with a threshold value t_x satisfying $0 < t_x \leq c_x$. For all leaf nodes x, $t_x = 1$. Given a set of attributes $\mathcal{A}$, $\mathcal{T}$ is evaluated from the leaves upward to R. A leaf is evaluated as 1, i.e. True, if and only if the corresponding attribute is enclosed in $\mathcal{A}$. Each inner node x is then evaluated as 1 if and only if at least t_x of its children nodes are evaluated as 1. If R is evaluated as 1, then $\mathcal{T}_R(\mathcal{A}) = 1$, namely, $\mathcal{A}$ matches access structure $\mathcal{T}_R$. An example access tree is depicted in Figure 6.1.

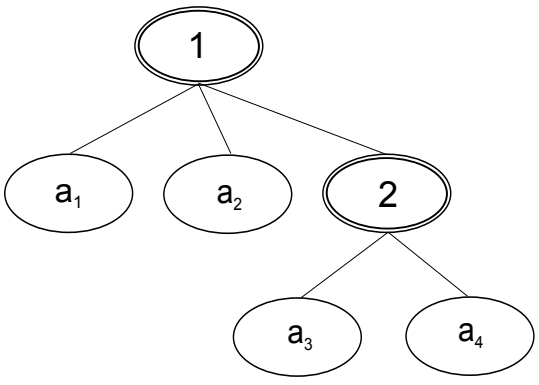


Figure 6.1: An access tree for the access structure $a_1 \vee a_2 \vee (a_3 \wedge a_4)$, where a_1, a_2, a_3, a_4 are four attributes.

The CP-ABE scheme in [7] consists of the following four algorithms:

AB-Setup is an initialization algorithm run by a trusted authority. It takes as input a security parameter, outputs a public key PK and a master secret key MK.

AB-KeyGen$(MK, \mathcal{A})$ is run by the authority to issue a key for a given attribute set $\mathcal{A}$. It takes as input MK and $\mathcal{A}$, outputs a key $SK_{\mathcal{A}}$ associated with $\mathcal{A}$.

AB-Encrypt($PK, m, \mathcal{T}$**)** is run by a user to perform a CP-ABE encryption on message m with an access structure $\mathcal{T}$. Taking as input PK, m and $\mathcal{T}$, it outputs a ciphertext $CT_{\mathcal{T}}$.

AB-Decrypt($CT_{\mathcal{T}}, \mathcal{T}, SK_{\mathcal{A}}, \mathcal{A}$**)** is run by a user holding $SK_{\mathcal{A}}$ to decrypt a ciphertext $CT_{\mathcal{T}}$ with a ciphertext policy $\mathcal{T}$. If $\mathcal{T}(\mathcal{A}) = 1$, it outputs the corresponding plaintext m correctly. Otherwise, it outputs $\perp$.

6.2 Access Control and Authentication for Generic Scalable Codestreams

As before, a source-proxy-user content distribution framework is considered and a scalable multimedia codestream with a set of layers $\{L_0, L_1, \cdots, L_m\}$, where L_0 is the base layer while $L_1, \cdots, L_m$ correspond to m enhancement layers is modeled. When joining the system, a user subscribes to a privilege level based on her preference. In general, the number of privilege levels is less than the number of layers. However, to simplify the presentation and without loss of generality, it is assumed that they are the same and that a user with privilege level j is entitled to access the set of layers $\{L_i\}_{i=0}^{j}$, $j = 0, 1, \cdots, m$.

The proposed scheme is constructed upon an authentication scheme in Chapter 3. To enforce access control on a codestream, the source picks a random root key k_m and then computes $k_{m-1} = \mathcal{H}(k_m), k_{m-2} = \mathcal{H}(k_{m-1}), \cdots, k_0 = \mathcal{H}(k_1), k_{MAC} = \mathcal{H}(k_0)$, where $\mathcal{H}(.)$ is a one-way hash function. The source uses k_{MAC} to authenticate the codestream as in the HMAC-based authentication scheme and uses k_i to

encrypt layer L_i, $i = 0, 1, \cdots, m$.

The CP-ABE scheme in subsection 6.1.1 is used to encrypt the encryption/access keys such that only privileged users can obtain the corresponding keys. For each k_j, $j = 0, 1, \cdots, m$, the source constructs a CP-ABE access tree Γ_j which has a single attribute node a_j corresponding to privilege level j with a threshold value 1. That is, $\Gamma_j(\{a_j\}) = 1$. The source computes the CP-ABE encryption on k_j with the single-node access structure Γ_j. Due to the simplicity of Γ_j, the resulting ciphertext has the minimum length. A user u subscribing to privilege level j holds the CP-ABE secret key SK_j^u associated with the attribute a_j and recovers k_j from the corresponding ciphertext. Then, starting from k_j, the user traverses the hash chain to get all the keys for decrypting the layers she is allowed to access.

The proposed access control and authentication scheme consists of four algorithms - *Initialization* which initializes the source and users' settings, *KeyDistribution* which distributes the symmetric encryption keys to users, *StreamGeneration* which encrypts and authenticates a multimedia codestream, and *StreamReceive* which decrypts and verifies a received codestream.

Initialization: The source runs **AB-Setup** to generate a CP-ABE PK and MK. When a user u registers to the source and subscribes to privilege level j, the source returns to her a $(SK_j^u, \mathcal{A}_j)$ tuple, where the attribute set $\mathcal{A}_j = \{a_j\}$ and $SK_j^u =$ **AB-KeyGen**$(MK, \mathcal{A}_j)$.

KeyDistribution: The source executes the following steps to generate and distribute symmetric keys to all users.

Step K1. Choose a random symmetric key k_m and generate $k_{m-1}, k_{m-2}, \cdots, k_0, k_{MAC}$ as described above.

Step K2. For each $j \in [0, m]$, construct an access structure Γ_j as specified above, and encrypt k_j to get $CT_j =$ **AB-Encrypt**(PK, k_j, Γ_j).

Step K3. Send the set of ciphertexts $\{CT_j, \Gamma_j\}_{j=0}^{m}$ to the users over either an in-band or out-of-band channel.

StreamGeneration: As the HMAC-based authentication scheme, the scheme here operates at the individual packet level. Given a packet $P = L_0 \| \cdots \| L_m$ in a multimedia codestream, the source performs the following steps.

Step A1. Generate an authenticated packet $P' = L'_0 \| \cdots \| L'_m$ by running the Authentication algorithm of an authentication scheme specified in Chapter 3 with the key k_{MAC}.

Step A2. Generate an encrypted and authenticated packet $P'' = L''_0 \| \cdots \| L''_m$ by computing $L''_i = Enc(L'_i, k_i)$, for all $0 \leq i \leq m$, where $Enc()$ is a symmetric encryption algorithm in a standard mode of operation, such as CBC or Counter mode [40].

Step A3. Output P'' as an authenticated and encrypted version of P.

StreamReceive: A user subscribing to privilege level j receives her granted keys during the key distribution phase. Specifically, when a user u with $(SK_j^u, \mathcal{A}_j)$ receives $\{CT_i, \Gamma_i\}_{i=0}^{m}$, she computes $k_j = \textbf{AB-Decrypt}(CT_j, \Gamma_j, SK_j^u, \mathcal{A}_j)$. Then, she traverses the hash chain to derive all the keys, $k_{j-1}, \cdots, k_0, k_{MAC}$, granted to her.

Upon receiving an encrypted and authenticated packet $P'' = L''_0 \| \cdots \| L''_{m-t}$ from a proxy, the user proceeds as follows.

Step V1. Decrypt L''_i to obtain $L'_i = Dec(L''_i, k_i)$, for $i = j, j-1, \cdots, 1, 0$.

Step V2. For each decrypted layer L'_j, execute the Verification algorithm in Section 3.3 using k_{MAC} as the verification key.

Note that incorporation of authentication in the encryption based access control scheme is not optional. It is well known that standard operation modes of block ciphers do not provide message authentication [40] and that using encryption without

adequate integrity protection is vulnerable to active attacks [43]. It is believed that integrity or authentication service must be offered in any security-aware transmission [41].

6.3 Access Control and Authentication for H.264/SVC Codestreams

In this section, the generic scheme presented in Section 6.2 is extended and applied to protect H.264/SVC codestreams. The description will focus on access control while omitting the part on codestream authentication. In addition, let y consecutive applications of the one-way function $\mathcal{H}(.)$ be denoted as $\mathcal{H}^y(x) = \mathcal{H}^{y-1}(\mathcal{H}(x))$.

In [93], an access control scheme is proposed for H.264/SVC-encoded video codestreams having temporal, spatial and quality scalabilities. A video codestream is modelled in two dimensions - the vertical dimension for spatial-quality layers of a frame and the horizontal dimension for different frames (i.e. temporal layers). In both dimensions, the higher layer depends on the lower layers for decoding. Let the term "unit", denoted as $D_{i,j}$, refer to spatial-quality layer i of a frame belonging to temporal layer j. Thus, a codestream with S spatial-quality layers and T temporal layers can have up to $S \times T$ units. The requirement of an access control scheme is such that for a user having access privilege to the unit $D_{s,t}$, the user will also have access to the set of units $\{D_{i,j} \mid i \in [0, s-1],\ j \in [0, t-1]\}$, but not the units higher than $D_{s,t}$ (see Figure 6.2).

For the sake of consistency, the scheme in [93] is illustrated in terms of the algorithms in Section 6.2 - namely *KeyDistribution*, *StreamGeneration* and *StreamReceive*.

KeyDistribution: Given an H.264 codestream as shown in Figure 6.2, the source chooses a random K, computes $k_Y = \mathcal{H}(K\|1)$ and $k_X = \mathcal{H}(K\|2)$, where k_Y and k_X denote the *scalability type keys* for the spatial-quality scalability and temporal

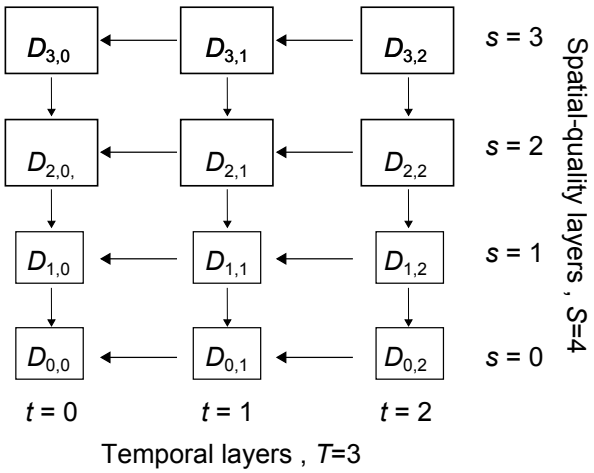


Figure 6.2: H.264/SVC-encoded codestream as modelled in [93] with $S = 4$ spatial-quality layers and $T = 3$ temporal layers. Direction of an arrow indicates a unit of the lower privilege, i.e. a privilege to access unit $D_{1,2}$ will also allow the user to access the units $D_{1,1}, D_{1,0}, D_{0,2}, D_{0,1},$ and $D_{0,0}$.

scalability, respectively. Then, the source computes, for all layers $i = S-1, S-2, \cdots, 0$ in the spatial-quality scalability, the set of keys $\{k_{Y,i}\}$ as $k_{Y,i} = \mathcal{H}^{S-i}(k_Y)$. Similarly, the source computes, for all layers $j = T-1, T-2, \cdots, 0$ in the temporal scalability, the set of keys $k_{X,j} = \mathcal{H}^{T-i}(k_X)$. It then forwards the keys securely to a KDC.

StreamGeneration: For every unit in spatial-quality layer i and temporal layer j, i.e. $D_{i,j}$, the source uses $k(i,j) = k_{Y,i} \| k_{X,j}$ as the symmetric key to encrypt $D_{i,j}$ and then send the encrypted codestream to the users.

StreamReceive: A user with access privilege to the unit $D_{s,t}$ first authenticates herself to the KDC to obtain the key $k(s,t) = k_{Y,s} \| k_{X,t}$. Using $k(s,t)$, the user derives all $k(p,q)$ for $p < s$ and $q < t$ and uses this set of keys to decrypt the units she is entitled to access.

The scheme in [93] allows both the KDC and a user to maintain only a single key, and a privileged user can derive all the necessary keys to decrypt the granted units using a one-way hash. However, the scheme is subject to collusion attack where two users separately subscribing to lower privilege levels can cooperate and

derive access key of a higher privilege level. With reference to Figure 6.2, suppose a user obtains $k(0,2) = k_{Y,0} \| k_{X,2}$ from the KDC and another user obtains $k(3,0) = k_{Y,3} \| k_{X,0}$. When they collude, they obtain $k(3,2) = k_{Y,3} \| k_{X,2}$ for decrypting the full codestream that they originally did not have privileges to access. This workaround is clearly unacceptable for most applications.

The following subsections show two approaches that are secure against this type of user collusion attack. In both approaches, the source classifies the layers into privilege levels. Note that a privilege level may cover one or more layers. However, to simplify explanation, it is assumed that each unit $D_{i,j}$ corresponds to a privilege level (i,j). As a result, there are $S \times T$ privilege levels: $(i,j), i \in [0, S-1],\ j \in [0, T-1]$. The source generates $S \times T$ access keys $k_{i,j}$ to encrypt $D_{i,j}$, $i \in [0, S-1],\ j \in [0, T-1]$, uses CP-ABE to encrypt $k_{i,j}$, and then sends the ciphertexts to users over either in-band or out-of-band channels. The two approaches differ in how the keys $k_{i,j}$ are generated.

6.3.1 Access Keys Generation - Approach 1

To generate a set of $S \times T$ access keys, the source

1. Generates a random key K.

2. For the highest temporal layer $T-1$ in every spatial-quality layer $i,\ i \in [0, S-1]$, computes $k_{i,T-1} = \mathcal{H}^{S-i}(K \| \text{“}S\text{”})$ where “S” is the ASCII code of the letter S.

3. In a given spatial-quality layer $s,\ s \in [0, S-1]$, for each of the remaining temporal layers $j,\ j \in [0, T-2]$, computes $k_{s,j} = \mathcal{H}^{T-1-j}(k_{s,T-1} \| \text{“}T\text{”})$ where “T” is the ASCII code of the letter T.

Figure 6.3 shows the keys generated for the codestream in Figure 6.2. Depending on a user's access requirements, there are two scenarios to be considered at the user end (see Figure 6.3).

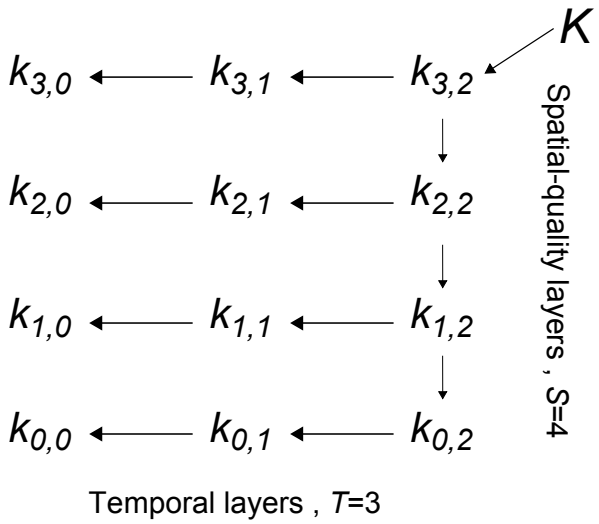


Figure 6.3: Access keys generated using Approach 1. If a user subscribes to privilege level $(2, 2)$, she needs to know the key $k_{2,2}$ to access the lower privilege levels; if the user subscribes to privilege level $(2, 1)$, she needs to know the keys $k_{2,1}, k_{1,1}, k_{0,1}$ in order to access the lower privilege levels.

1. User subscribes to privilege level $(s, T-1)$ (i. e., to a codestream with certain spatial-quality layer s but with the highest temporal layer $T-1$). The user only needs to obtain $k_{s,T-1}$ from the corresponding CP-ABE ciphertext. Using $k_{s,T-1}$, she can derive the set of keys $\{k_{i,j}\}$ for $i \in [0, s-1]$ and $j \in [0, T-2]$.

2. User subscribes to privilege level (s, t) for some spatial-quality layer s and some temporal layer t. The user needs to first obtain $\{k_{i,t}\}_{i\in[0,s]}$ by decrypting $s+1$ CP-ABE ciphertexts and then computes the other keys necessary for decrypting the units she is entitled to access.

This approach allows the source to maintain a single key but a user has to maintain potentially more than one keys due to the sequential key generation. The efficiency of this approach is the highest if a user subscribes to a codestream of the highest temporal layer (regardless of which spatial-quality layer) since she needs to perform only one CP-ABE decryption. On the other hand, if the user subscribes to a temporal layer other than the highest layer, she needs to perform more than one CP-ABE decryptions - the maximum being the number of available spatial-quality

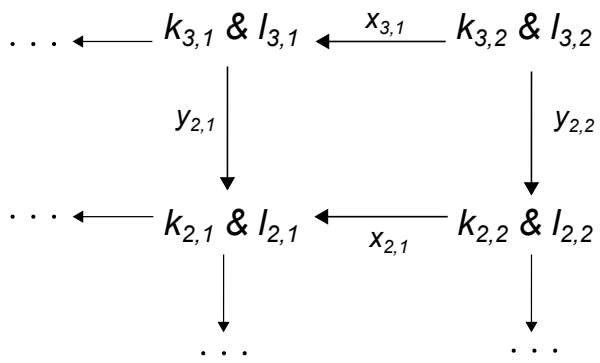


Figure 6.4: Keys generated using Approach 2, where $x_{3,1} = k_{3,1} \oplus \mathcal{H}(k_{3,2} \| l_{3,1})$; $y_{2,2} = k_{2,2} \oplus \mathcal{H}(k_{3,2} \| l_{2,2})$; $x_{2,1} = k_{2,1} \oplus \mathcal{H}(k_{2,2} \| l_{2,1})$; $y_{2,1} = k_{2,1} \oplus \mathcal{H}(k_{3,1} \| l_{2,1})$.

layers in the codestream[1].

6.3.2 Access Keys Generation - Approach 2

In the second approach, a key generation method proposed in [4] is utilized. To generate a set of $S \times T$ access keys, for each privilege level (i, j), the source chooses a random secret access key $k_{i,j}$, a unique public label $l_{i,j}$ and where applicable, public values $y_{i-1,j} = k_{i-1,j} \oplus \mathcal{H}(k_{i,j} \| l_{i-1,j})$ and $x_{i,j-1} = k_{i,j-1} \oplus \mathcal{H}(k_{i,j} \| l_{i,j-1})$ where $\oplus$ is an exclusive OR operation.[2]

Figure 6.4 shows the keys generated for the codestream in Figure 6.2. It is stressed that the secret keys $k_{i,j}$ are CP-ABE encrypted. The CP-ABE ciphertexts and the public values (i.e., $l_{i,j}$, $y_{i-1,j}$ and $x_{i,j-1}$) are sent to users via either an in-band or out-of-band channel.

Note that a user subscribing to a privilege level (s, t) can always derive keys for the lower privilege levels using the public information and the single key $k_{s,t}$ as follows. The user first computes $k_{s-1,t} = y_{s-1,t} \oplus \mathcal{H}(k_{s,t} \| l_{s-1,t})$ and $k_{s,t-1} = x_{s,t-1} \oplus \mathcal{H}(k_{s,t} \| l_{s,t-1})$, and then computes all $k_{i,j}$ for $i \in [0, s-2]$ and $j \in [0, t-2]$.

This approach similarly eliminates the need of an online KDC, and requires the source and users to maintain only a single key. As a result, it allows each user

[1] Note that the source can always alter the key generation sequence depending on user request pattern to achieve the optimal efficiency.

[2] $|k_{i,j}| = |l_{i,j}| = |y_{i,j}| = |x_{i,j}| = |\mathcal{H}(.)|$.

to perform the minimum number of one CP-ABE decryption. It is secure against collusion attack due to the one-way function $\mathcal{H}(.)$ and it allows users of higher privilege to efficiently derive access keys for the lower privileges but not vice versa. However, the public values $l_{i,j}$, $y_{i-1,j}$ and $x_{i,j-1}$ must be delivered to users which results in higher communication overhead than Approach 1.

6.4 Remarks and Discussions

The access control schemes in [106] and [105] are designed for the MPEG-4 FGS codestreams and involve an online KDC. As mentioned, the scheme in [106] generates independent keys for different privilege levels. As a result, both the KDC and users have to maintain a large number of keys for every video codestream. This number is then reduced to one per video codestream in [105] but this scheme is vulnerable to the same user collusion attack as described in Section 6.3. Note that the MPEG-4 FGS and the H.264/SVC codestream structures are similar. Hence, the proposed approaches for H.264/SVC codestreams can be readily applied for access control of MPEG-4 FGS codestreams as well without the need for an online KDC and without suffering from the user collusion attack.

Users may subscribe and unsubscribe from a multimedia service. In the case that a new user subscribes to a multimedia service at a specific privilege level, the source issues to her a CP-ABE secret key associated with that privilege level. This is an one time effort and can be done either online or offline. Encrypted multimedia codestreams can be broadcast to users, and only those users with the secret keys at the right privilege levels are able to successfully decrypt the received codestreams. Whenever a user terminates the subscription, the source must "revoke" the user such that she can no longer access the multimedia service. As pointed out in [7], this can be achieved by incorporating numerical attributes in a user's CP-ABE secret key. For instance, when a user subscribes to a certain privilege level, the source provides

the user a CP-ABE secret key associated with an attribute specifying an expiry date[3]. Before the expiry date, the user will be able to access multimedia codestreams at her access privilege. Once the time lapses, the user will need to obtain a new CP-ABE secret key with a new expiry date. Note that for better security, it is prudent practice to encrypt each video codestream using a different secret key; otherwise compromising one key will result in compromising multiple video codestreams. Without employing CP-ABE, an online KDC is needed to distribute these secret keys to end users over authenticated and secure channels. The KDC would be operated by the source or other parties. In any case, an online KDC leads to higher operating cost and, if not managed properly, could be a single point of failure.

An access control scheme that allows flexible privilege classifications for generic scalable multimedia codestreams is proposed. The schemes use CP-ABE to distribute access keys to users, thereby eliminating the need of an online KDC. The scheme is also extended for access control of H.264/SVC codestreams. In addition, a user collusion attack to the existing multimedia codestreams access control schemes in the literature is pointed out, and two key generation techniques that are secure against the collusion attack are presented.

[3]Such application is feasible since most multimedia subscription is on a monthly or yearly basis.

Chapter 7

Conclusion and Future Work

This dissertation makes contributions on the issue of authentication and access control in online distribution of multimedia codestreams. For authentication, the cryptographic-based and content-based authentication solutions are studied, respectively, as a mean to authenticate multimedia codestreams in the event of transcoding. Two cryptographic-based authentication schemes for generic scalable codestreams arr presented, combining the hash-chaining technique and double error correction coding algorithm to achieve proxy-transparency and resiliency to packet losses with a low communication overhead. To reduce computation cost, the second scheme replaces digital signature with a hash-based message authentication code (HMAC) and the HMAC-based authentication scheme achieves a packet-level loss-resiliency. The performance of the schemes are analyzed and it is shown that although the signature-based scheme has a higher communication overhead compared to existing schemes, they have similar computation and verification times while the signature-based scheme possesses the proxy-transparency property. The HMAC-based authentication scheme further outperforms the signature-based scheme in terms of computation time, verification delay and loss-resiliency.

The signature-based scheme for generic codestreams is further extended to address the issue of authentication for H.264/SVC codestreams, where in addition to spatial-quality layers within a frame, an H.264/SVC codestream is also encoded

with one or more temporal layers, each containing a subset of frames. By integrating the temporal scalability structure with double error correction coding and packet replications, the scheme is robust to transcoding and it achieves a high loss-resiliency with a low communication overhead under burst loss condition. The proposed scheme is further compared and analyzed with an existing scheme in [56] in terms of its loss-resiliency under different conditions such as the spatial-quality layer dependency structures and codestream transmission orders.

As content-based authentication solution is commonly used to authenticate non-scalable codestreams in the event of transcoding, a study on the security of existing transform-domain content-based authentication schemes is performed. Based upon the concept of video coding, a common design flaw in the existing schemes, where the transform-domain feature extracted by these schemes cannot truly represent the codestream semantic, is pointed out. As a consequent, an attacker is able to manipulate transform-domain parameters to mount semantic-changing attacks that cannot be detected by the schemes. A discussion on how the flaw can be exploited is presented and several attack examples are shown. A further analysis on the attacks that manipulate transform-domain header parameters, and the conditions required for the attacks to succeed given an attacker's desired content is presented.

Finally, the issue of access control for H.264/SVC codestreams for applications such as video-on-demand is addressed. The proposed access control scheme uses symmetric encryption to encrypt each spatial-quality/temporal layer with a different access key and uses ciphertext-policy attribute-based encryption as a mean to disseminate access keys to authorized users; as a result, the scheme eliminates the need to deploy an online key distribution center, which could pose a scalability problem as the number of users increases. Using a secure and efficient access keys generation hierarchy that is fully compatible with the H.264/SVC codestream structure, the proposed access control scheme is secure against collusion attack, and is efficient in the sense that a user needs to only maintain a single access key, regardless of the privilege level he/she subscribes to.

7.1 Future Directions

One of the future directions is to design a secure content-based authentication schemes for non-scalable codestreams in the event of transcoding. Non-scalable codestreams are commonly transcoded by re-encoding the codestream using a new quantization step size. However, the process of full decoding and re-encoding is computationally intensive. Different transcoding techniques have then been proposed to speed up the transcoding process without compromising the visual presentation of transcoded codestreams. More notably, it is shown in [15] that intra-predicted frames should be transcoded by a full-decode-and-reencode process to eliminate drift errors in the transcoded codestream, whereas inter-predicted frames can be transcoded by reusing header parameters and performing motion compensation in the transform domain. The aim is to design a content-based authentication scheme that can extract a secure and robust feature that can survive the different transcoding techniques for intra- and inter-predicted frames.

One of the challenges in designing a content-based authentication scheme for scalable codestreams is the robustness of the feature. A feature that is extracted from a low resolution frame is unable to correctly verify the content of a high resolution frame due to the lower entropy, and vice versa. Most existing content-based solutions require the verifier to downsample a high resolution frame before verification. The aim is to eliminate this limitation by studying the characteristics of transform-domain parameters of a scalable codestream, as well as the significance of signaling parameters for decoding. Thus, by extracting important feature from the base layer of the codestream and from important transform-domain and signaling parameters from the enhancement layers, the proposed scheme should be secure against malicious tampering while remain robust to transcoding.

The Multiview Video Coding (MVC) standard [87] is a new standard for video compression that allows efficient encoding of video sequences captured simultaneously from multiple camera angles in a single codestream. MVC codestreams

are backward compatible with H.264/AVC, where an H.264/AVC decoder can decode MVC codestreams while ignoring the second view codestream. Apart from its primary usage for 3D display systems, an MVC codestream can be used for free-viewpoint videos [78], where users can navigate through different viewpoints of a scene - whether it is commercial (sports scenes and surveillance) or immersive teleconference applications. The aim is to study the coding structure of an MVC codestream and design an efficient authentication scheme that is fully compatible with this structure to achieve one-time authentication while allowing authenticity verification of different views.

Bibliography

[1] American Hospital Association. *The promise of telehealth for hospitals, health systems and their communities*. TrendWatch, January 2015.

[2] Apple ProRes White Paper. Apple ProRes. Available online at: https://www.apple.com/final-cut-pro/docs/Apple_ProRes_White_Paper.pdf. June 2014.

[3] Arizona State University. Video trace library. Available online at: http://trace.eas.asu.edu/index.html. 19 March 2015.

[4] M. J. Atallah, K. B. Frikken, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pages 190–202, 2005.

[5] G. Ateniese, D. H. Chou, B. Medeiros, and G. Tsudik. Sanitizable signatures. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS)*, volume 3679, pages 159–177, 2005.

[6] BBC. Greece debt: Confusion in Varoufakis middle finger row. Available online at: http://www.bbc.com/news/world-europe-31961254. 19 March 2015.

[7] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, pages 321–334, 2007.

[8] T. Bianchi, A. D. Rosa, and A. Piva. Improved DCT coefficient analysis for forgery localization in JPEG images. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447, 2011.

[9] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[10] C. Chang, Y. Fan, and W. Tai. Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognition*, 41(2):654–661, 2008.

[11] Cisco VNI Report. *The Zettabyte Era: Trends and Analysis*. Cisco White Paper, May 2015.

[12] CMO Council. *The role of visual media in impactful brand storytelling*. CMO Council White Paper, August 2015.

[13] W. Dai. Crypto++ 5.5 benchmarking. Available online at: http://www.cryptopp.com/benchmarks.html. Accessed on: 21 March 2011.

[14] Y. Dai, S. Thiemert, and M. Steinebach. Feature-based watermarking scheme for MPEG-I/II video authentication. In *Proceedings of SPIE Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 325–335, 2004.

[15] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle. Requantization transcoding for H.264/AVC video coding. *Signal Processing: Image Communication*, 25:235–254, January 2010.

[16] R. H. Deng and Y. Yang. A study of content authentication in proxy-enabled multimedia delivery systems: Model, techniques, and applications. *ACM Transactions on Multimedia Computing , Communications and Applications*, 5(4):28:1–28:20, 2009.

[17] J. Dittmann, A. Steinmetz, and R. Steinmetz. Content-based digital signature for motion pictures authentication and content-fragile watermarking. In *Proceedings of the 1999 IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 209–213, 1999.

[18] R. Du and J. Fridrich. Lossless authentication of MPEG-2 video. In *Proceedings of the 2002 International Conference on Image Processing (ICIP)*, volume 2, pages II–893–II–896, 2002.

[19] T. Ebrahimi and C. Horne. MPEG-4 natural video coding - An overview. *Signal Processing: Image Communication*, 15(4-5):365–385, 2000.

[20] eMarketer. Photos cluterring your Facebook feed? Here's why. Available online at: http://emarketer.com/Article/Photos-Cluttering-Your-Facebook-Feed-Herersquos-Why/1010777. April 21, 2014.

[21] Ericsson White Paper. Understanding ultra high definition television. Available online at: http://www.ericsson.com/res/docs/whitepapers/wp-uhd.pdf. November 2015.

[22] Extron Electronics White Paper. Hitting the moving target of 4K. Available online at: http://www.extron.com/download/files/whitepaper/extron4k_wp.pdf. April 17, 2015.

[23] Federal Information Processing Standard Publications. Secure hash signature standard (SHS). FIPS PUB 180-2, National Institute of Standards and Technology, August 2002.

[24] P. Ferré, D. Agrafiotis, T. K. Chiew, A. Doufexi, A. Nix, and D. Bull. Packet loss modelling for H.264 video transmission over IEEE 802.11g wireless LANs. In *IEEE 13th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2005.

[25] J. Fridrich. Security of fragile authentication watermarks with localization. In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents IV*, volume 4675, pages 691–700, 2002.

[26] R. Ganot. Video marketing statistics for 2015: The next big thing is here. Available online at: http://www.codefuel.com/blog/video-marketing-statistics-for-2015-the-next-big-thing-is-here/. May 7, 2015.

[27] Geeknet Inc. Open SVC decoder. Available online at: http://sourceforge.net/apps/mediawiki/opensvcdecoder/index.php?title=Main_Page. Accessed on: 12 March 2012.

[28] R. Gennaro and P. Rohatgi. How to sign digital streams. In *Advances in Cryptology: Proceedings of CRYPTO '97*, volume 1294, pages 180–197, 1997.

[29] C. Gentry, A. Hevia, R. Jain, T. Kawahara, and Z. Ramzan. End-to-end security in the presence of intelligent data adapting proxies: The case of authenticating transcoded streaming media. *IEEE Journal on Selected Areas in Communications*, 23(2):464–473, 2005.

[30] E. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:1253–1265, 1960.

[31] P. Golle and N. Modadugu. Authenticating streamed data in the presence of random packet loss. In *Proceedings of the 8th Annual Network and Distributed Systems Security Symposium (NDSS)*, pages 13–22, 2001.

[32] M. Hefeeda and K. Mokhtarian. Authentication schemes for multimedia streams: Quantitative analysis and comparison. *ACM Transactions on Multimedia Computing, Communications and Applications*, 6(1):1–24, February 2010.

[33] M. Hoelzel and M. Ballvé. *The Programmatic-Advertising Report: Mobile, video and real-time bidding drive growth in programmatic*. BI Intelligence Report, March 26, 2015.

[34] M. Holliman and N. Memon. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Transactions on Image Processing*, 9(3):432–442, 2000.

[35] S.-J. Horng, M. E. Farfoura, P. Fan, X. Wang, T. Li, and J.-M. Guo. A low cost fragile watermarking scheme in H.264/AVC compressed domain. *Multimedia Tools and Applications*, pages 1–27, 2013.

[36] H.-T. Hu and L.-Y. Hsu. Exploring DWT-SVD-DCT feature parameters for robust multiple watermarking against JPEG and JPEG2000 compression. *Computers and Electrical Engineering*, 41(0):52–63, 2015.

[37] ISO/IEC JTC 1/SC 29. Information technology - Generic coding of moving pictures and associated audio information - Part 2: Video. ISO/IEC 13818-2:2013, ISO/IEC, October 2013.

[38] J. Worland (TIME). Sandra bland arrest video appears edited. Available online at: http://time.com/3967329/sandra-bland-video-continuity/. July 22, 2015.

[39] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6). *JM Reference Software Manual*, 2009.

[40] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2008.

[41] S. Kent. IP encapsulating security payload (ESP). RFC 4303, The Internet Society, December 2005.

[42] D. Kim, Y. Choi, H. Kim, J. Yoo, H. Choi, and Y. Seo. The problems in digital watermarking into intra-frames of H.264/AVC. *Image Vision Computing*, 28(8):1220–1228, January 2010.

[43] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In *Advances in Cryptology: Proceedings of CRYPTO 2001*, pages 310–331, 2001.

[44] T.-Y. Kuo, Y.-C. Lo, and C.-I. Lin. Fragile video watermarking technique by motion field embedding with rate-distortion minimization. In *Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, pages 853–856, 2008.

[45] C.-Y. Lin and S.-F. Chang. Issues and solutions for authenticating MPEG video. In *Security and Watermarking of Multimedia Contents*, 1999.

[46] C. Y. Lin and S. F. Chang. Semi-fragile watermarking for authenticating JPEG visual content. In *Proceedings of the SPIE Security and Watermarking of Multimedia Contents*, pages 140–151, 2000.

[47] C. Y. Lin and S. F. Chang. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and Sysems for Video Technology*, 11(2):153–168, 2001.

[48] P. Lin, C. Hsieh, and P. Huang. A hierarchical digital watermarking method for image tamper detection and recovery. *Pattern Recognition.*, 38(12):2519–2529, 2005.

[49] S. Lin and J. D. J. Costello. *Error Control Coding: Fundamentals and Applications, 2nd ed.* NJ: Prentice-Hall, 2004.

[50] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[51] Z. Luo, L. Song, S. Zheng, and N. Ling. H.264 advanced video control perceptual optimization coding based on JND-directed coefficient suppression. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(6):935–948, June 2013.

[52] S. Ma, W. Gao, D. Zhao, and Y. Lu. A study on the quantization scheme in H.264/AVC and its application to rate control. In *Advances in Multimedia Information Processing, PCM*, pages 192–199, 2004.

[53] A. Mansouri, A. M. Aznaveh, F. Torkamani-Azar, and F. Kurugollu. A low complexity video watermarking in H.264 compressed domain. *IEEE Transactions on Information Forensics and Security*, 5(4):649–657, December 2010.

[54] S. Miner and J. Staddon. Graph-based authentication of digital streams. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy (S&P)*, pages 232–246, 2001.

[55] B. G. Mobasseri and Y. J. NaikRaikar. Authentication of H.264 streams by direct watermarking of CAVLC blocks. In *Proceedings of the SPIE security, steganography, and watermarking of multimedia contents VIII*, volume 6505, pages 1–5, 2007.

[56] K. Mokhtarian and M. Hefeeda. End-to-end secure delivery of scalable video streams. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 09)*, pages 79–84, 2009.

[57] M. Naccari and F. Pereira. Advanced H.264/AVC-based perceptual video coding: Architecture, tools and assessment. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(6):806–819, June 2011.

[58] Network Systems Lab. svcAuth - NSL. Available online at: http://nsl.cs.sfu.ca/wiki/index.php/svcAuth. Accessed on: April 1, 2012.

[59] Offspark.com. Polar SSL. Available online at: http://polarssl.org/. Accessed on: 12 March 2012.

[60] A. Pannetrat and R. Molva. Efficient multicast packet authentication. In *Proceedings of the 10th Annual Network and Distributed Systems Security Symposium (NDSS)*, 2003.

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

[62] J. S. Park and H. J. Song. Selective intra prediction mode decision for H.264/AVC encoders. In *World Academy of Science, Engineering and Technology 13*, pages 51–55, 2006.

[63] S.-W. Park and S. Shin. Authentication and copyright protection scheme for H.264/AVC and SVC. *Journal of Information Sciences and Engineering*, 27:129–142, 2011.

[64] A. Perrig, R. Canetti, J. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy (S&P)*, pages 56–73, 2000.

[65] J. S. Plank, S. Simmerman, and C. D. Schuman. Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2. Technical Report CS-08-627, University of Tennessee, August 2008.

[66] C. I. Podilchuk and E. J. Delp. Digital watermarking: Algorithms and applications. *Signal Processing Magazine, IEEE*, 18(4), 2001.

[67] A. Rajput. *Benefits of network video for retail: A new perspective on retail surveillance*. Axis Communications White Paper, March 26, 2015.

[68] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, 8(2):300–304, 1960.

[69] J. Reichel, H. Schwarz, M. Wien, and J. Vieron. Joint scalable video model 9 of ISO/IEC 14496-10:2005/AMD3 Scalable Video Coding. Joint Video Team (JVT) X202, ISO/IEC/AMD3, 2007.

[70] Y. J. Ren, L. O'Gorman, L. J. Wu, F. Chang, T. L. Wood, and J. R. Zhang. Authenticating lossy surveillance video. *IEEE Transactions on Information Forensics and Security*, 8(10):1678–1687, 2013.

[71] I. E. Richardson. *The H.264 Advanced Video Compression Standard – 2nd edition*. John Wiley and Sons, Ltd., 2010.

[72] H. Rifà-Pous and J. Herrera-Joancomartí. Computational and energy costs of cryptographic algorithms on handheld devices. *Future Internet*, 3(1):31–48, 2011.

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

[74] K. A. Saadi, A. Bouridane, and A. Guessoum. Combined fragile watermark and digital signature for H.264/AVC video authentication. In *Proceedings of the 17th European Signal Processing Conference*, pages 1799–1803, 2009.

[75] H. Schwarz, D. Merpe, 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.

[76] S. Shahabuddin, R. Iqbal, S. Shirmohammadi, and J. Zhao. Compressed-domain temporal adaptation-resilient watermarking for H.264 video authentication. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1752–1755, 2009.

[77] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, September 2001.

[78] A. Smolic and P. Kauff. Interactive 3-D video representation and coding technologies. *Proceedings of the IEEE*, 93(1):98–110, January 2005.

[79] D. Song, D. Zuckerman, and J. D. Tygar. Expander graphs for digital stream aauthentication and robust overlay networks. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P)*, pages 258–270, 2002.

[80] P.-C. Su, C.-C. Chen, and H.-M. Chang. Towards effective content authentication for digital videos by employing feature extraction and quantization. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(5):668–677, 2009.

[81] Q. Sun, D. He, and Q. Tian. A secure and robust authentication scheme for video transcoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(10):1232–1244, October 2006.

[82] W.-L. Tang and S.-H. Yang. Optimal GOP size of H.264/AVC temporal scalable coding. *Advances in Intelligent Systems and Applications*, 2:403–412, 2012.

[83] TecHome. A look ahead to the top home tech trends of 2015. Available online at: http://techomebuilder.com/emagazine-articles/a-look-ahead-to-the-top-home-tech-trends-of-2015/. December 29, 2014.

[84] Telecommunication Standardization Sector of ITU. Advanced video coding for generic audiovisual services. ITU-T H.264 ISO/IEC 14496-10, International Telecommunication Union, February 2014.

[85] The CAVIAR team. EC funded CAVIAR project/IST 2001 37540. Available online at: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[86] G. C. Ting, B. M. Goi, and S. W. Lee. Cryptanalysis of a fragile watermark based H.264/AVC video authentication scheme. *Applied Mechanics and Materials*, 145:552–556, 2011.

[87] A. Vetro, T. Wiegand, and G. J. Sullivan. Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard. *Proceedings of the IEEE*, 99(4):626–642, April 2011.

[88] Y. Wang and A. Pearmain. Blind MPEG-2 video watermarking robust against geometric attacks: A set of approaches in DCT domain. *IEEE Transactions on Image Processing*, 15(6):1536–1543, June 2006.

[89] Z. Wei, Y. Wu, R. Deng, and X. Ding. A hybrid scheme for authenticating scalable video codestreams. *IEEE Transactions on Information Forensics and Security*, 9(4):543–553, April 2014.

[90] S. Wenger, Y. K. Wang, T. Schierl, and A. Eleftheriadis. RTP payload format for scalable video coding. RFC 6190, Internet Engineering Task Force (IETF), May 2011.

[91] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):53–61, July 2003.

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

[93] Y. Won, T. Bae, and Y. Ro. Scalable protection and access control in full scalable video coding. In *Proceedings of the 5th International Workshop on Digital Watermarking (IWDW)*, volume 4283, pages 407–421, 2006.

[94] C. Wong and S. Lam. Digital signatures for flows and multicasts. *IEEE Transactions on Networking*, 7(4):502–513, 1999.

[95] P. Wong. A watermark for image integrity and ownership verification. In *Proceedings of the IS and TS PICS Conference*, pages 374–379, 1998.

[96] Y. Wu and R. H. Deng. Scalable authentication of MPEG-4 streams. *IEEE Transactions on Multimedia*, 8(1):152–161, 2006.

[97] Y. Wu and C. Xu. A fault-induced attack to semi-fragile image authentiation schemes. In *Proceedings of SPIE on Visual Communications and Image Processing (vol. 5150)*, pages 1875–1883, 2003.

[98] X. Yang, W. Lin, Z. Lu, E. Ong, and S. Yao. Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(6):742–752, June 2005.

[99] M. Yeung and F. Mintzer. An invisible watermarking technique for image verification. In *Proceedings of the 1997 International Conference on Image Processsing (ICIP)*, pages 680–683, 1997.

[100] P. Yin and H. H. Yu. A semi-fragile watermarking system for MPEG video authentication. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–3461–IV–3464, 2002.

[101] H. H. Yu. A loss resilient and scalable streaming media authentication scheme. In *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference*, pages 60–64, 2005.

[102] W. Zhang, R. Zhang, X. Liu, C. Wu, and X. Niu. A video watermarking algorithm of H.264/AVC for content authentication. *Journal of Networks*, 7(8):1150–1154, August 2012.

[103] Z. Zhang, Q. Sun, and W.-C. Wong. A proposal of butterfly-graph based stream authentication over lossy networks. In *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME)*, page 4 pages, 2005.

[104] Z. Zhao and P. Liang. A statistical analysis of H.264/AVC FME mode reduction. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(1):53–61, January 2011.

[105] B. B. Zhu, M. Feng, and S. Li. An efficient key scheme for layered access control of MPEG-4 FGS video. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 443–446, 2004.

[106] B. B. Zhu, C. Yuan, Y. Wang, and S. Li. Scalable protection for MPEG-4 fine granularity scalability. *IEEE Transactions on Multimedia*, 7(2):222–233, 2005.

[107] D. K. Zou and J. A. Bloom. H.264/AVC stream replacement technique for video watermarking. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1749–1752, 2008.

[108] D. K. Zou and J. A. Bloom. H.264 stream replacement watermarking with CABAC encoding. In *Proceedings of the 2010 IEEE International Conference on Multimedia and Expo (ICME)*, pages 117–121, 2010.