Quality-Optimized and Secure End-to-End Authentication for Media Delivery

Qibin Sun, *Member, IEEE,* John Apostolopoulos, *Senior Member, IEEE,* Chang Wen Chen, *Fellow, IEEE,* and Shih-Fu Chang, *Fellow, IEEE*

***Abstract*—The need for security services, such as confidentiality and authentication, have become one of the major concerns in multimedia communication applications, such as video on demand and peer-to-peer content delivery. Conventional data authentication can not be directly applied for streaming media when an unreliable channel is used and packet loss may occur. This paper begins by reviewing existing end-to-end media authentication schemes, which can be classified into stream-based and content-based techniques. We then motivate and describe how to design authentication schemes for multimedia delivery which exploit the unequal importance of different packets. By applying conventional cryptographic hashes and digital signatures to the media packets, the system security is similar to that achievable in conventional data security. However, instead of optimizing packet verification probability, we optimize the quality of the authenticated media, which is determined by the packets which are received, decodable, and authenticatable. The quality of the authenticated media is optimized by allocating the authentication resources unequally across streamed packets based on their relative importance, thereby providing unequal authenticity protection (UAP). The effectiveness of this approach is demonstrated through experimental results on different media types (image and video), different compression standards (JPEG, JPEG2000, and H.264) and different channels (wired with packet erasures and wireless with bit errors).**

***Index Terms*—media authentication, media security, streaming media authentication, stream authentication, video streaming**

I. Introduction

MEDIA communication over heterogeneous networks is continuing to increase in practical importance, enabled by the rapid growth of network bandwidth, improved compression formats [1], and advanced delivery technologies such as content delivery networks [2] and peer-to-peer systems [3][4][5]. This is also evident in many commercial services and applications like IPTV (i.e., Internet Protocol Television), multimedia messaging, video conferencing, video surveillance, and so on. However, security issues such as confidentiality, authentication and secure media adaptation [6][7] are also becoming serious concerns. For instance, the content sender wants to ensure that his content can only be viewed by authorized people, and the content viewer also wants to ensure that the received content is indeed from the right sender and that it has not been accidentally or maliciously altered. Confidentiality of the content, to limit user's access, which is achieved by encryption has received considerable attention in recent years [8][9][10][11]. In this paper we examine the problem of authentication for media delivery. While data authentication is well understood and many practical solutions exist, authentication for streaming media is challenging because the media delivery is often over an unreliable channel where packet loss may occur. Specifically, when streaming over a lossy packet network individual packets in the stream could be lost or modified during transmission.

Media authentication is a relatively new research area compared to other traditional research areas such as multimedia compression. Researchers in different areas and with different technical backgrounds may use different definitions for media "authentication". For example, the biometrics community may use the term authentication to mean source (e.g., face, fingerprint) identification or verification; the multimedia watermarking community usually uses the term authentication to refer to content integrity protection; Note that content integrity protection based on digital watermarking [12][13] is another attractive research area with many potential applications such as video surveillance [14].

Throughout this paper, we define *authentication* as the process by which the authorized receivers, and perhaps the arbiters, determine whether a specified data has, with very high probability, been (1) sent by the authorized sender and (2) has not been subsequently altered or substituted for [15] (page. 382). In other words, authentication will answer two questions: (1) who sent the data (non-repudiation), and 2) whether the received data has been altered or not (data integrity). Therefore, in this paper the term *authentication* means both source and data authentication. To maintain the security performance at a computationally infeasible level for potential attackers, the above definition usually requires that the received media must be identical to what was sent, as in the case of conventional data authentication when the data is delivered over a reliable channel or transport protocol. However, this is not the case when streaming media over a lossy network.

The problem context is illustrated in Figure 1. The original image is coded and packetized, using error resilient techniques, for delivery over the lossy channel. We assume that the transmission channel is neither reliable nor secure, where some data packets may be naturally lost (e.g., due to congestion) or maliciously modified before reaching the receiver. At the receiver, the corrupted image can be approximately recovered by error concealment techniques before displaying or further processing.

From Fig.1, we see that the typical requirement assumed for data authentication, that the data at the receiver (either the received coded media or the decoded media) is the exact same as what was sent by the sender, is not appropriate because the network loss would make the received coded media incomplete and the decoded media would have degraded quality. In this context, a more useful definition is that the authenticated media corresponds to the media decoded exclusively from authenticated packets. This definition prevents packet alteration (note that it may not be possible to identify whether a packet loss is accidental or malicious). Under this definition, a packet is consumed only when it is received, decodable, and authenticated. Therefore, in this paper, for authentication of streaming media, we use the following two guiding principles. First, even if a received media stream is incomplete, the goal is to still try to authenticate all the received packets. Ideally, every received packet can be authenticated. Second, a received media packet is consumed only when it is both decodable and authenticated. A received and decodable but unauthenticated packet should not be consumed because of security concerns. Similarly, an authenticated but undecodable packet is also useless. Therefore, ideally each packet would be independently decodable and independently authenticatable.

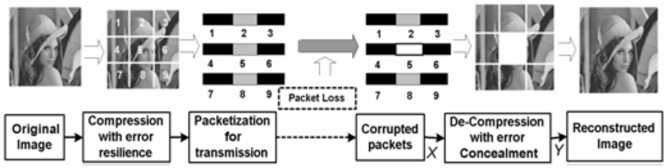

1
2
3
4
5
6
7
8
9
Packet Loss
Original Image
Compression with error resilience
Packetization for transmission
Corrupted packets
X
De-Compression with error Concealment
Y
Reconstructed Image

Fig.1. Image delivery over unreliable channel

In Figure 1, at the receiver authentication can be performed at either point *X* or point *Y*, depending on the application. These two points classify existing authentication approaches into two classes: (1) stream-based authentication (authenticating media data packets at point *X*), and (2) content-based authentication (authenticating media content at point *Y*). More detailed introductions on existing solutions are given in the next section, and here we only highlight their high-level attributes. Stream-based methods have the advantage that they provide a similar level of security to conventional data security techniques, and very importantly provide mathematically provable levels of security. Their disadvantages include that the extra bit rate overhead can be significant, computational complexity can be high, and the quality of the authenticated media can be far inferior to the quality of the same media afflicted by the same losses, but without requiring authentication. On the other hand, content-based methods, which are typically achieved via some form of digital watermarking, generally require less bit rate overhead and they are usually more robust to media distortions. However, it is generally much more difficult to make useful and mathematically provable statements about the system security for content-based methods, and generally the level of security is significantly less.

The above mentioned limitations of the conventional media authentication approaches motivates us to revisit stream-based methods and study whether the quality of the received media can be optimized using information from the media content. Further motivation for exploring this direction is based on the following observations. First, being a special type of data, different media packets typically have different importance depending on the compression and media content. Therefore, it is a natural idea to allocate more authentication resources toward the more important packets. Second, media is usually coded according to certain compression standards before streaming, and this leads to coding dependencies between the different packets. These dependences should also be taken into consideration for resource allocation. Lastly but most importantly, while previous stream-based authentication techniques aim to optimize the authentication (i.e., verification) probability of individual packets, the goal of media streaming is generally to maximize the media quality provided to the end-user. Therefore, media quality is arguably a more important metric than verification probability for evaluating and optimizing the performance of streaming media authentication techniques.

This paper focuses on application-layer end-to-end authentication, as opposed to transport layer such as Secure Sockets Layer (SSL) / Transport Layer Security (TLS) [16], or network layer techniques such as Internet Protocol Security (IPSec) [17]. SSL/TLS and IPSec all provide authentication capabilities. IPSec operates at the network layer enabling authentication of each individual packet, while SSL/TLS operates at the transport layer authenticating each message comprising potentially many packets. SSL/TLS operates on top of Transmission Control Protocol (TCP) which provides a reliable connection (no packet loss). Comparing to SSL/TLS, the proposed approach is robust to packet loss and therefore does not need a reliable connection (TCP). Note that TCP is unacceptable for many media streaming applications because of the large delays that often result from its persistent retransmissions, and because the abrupt changes of its additive-increase multiplicative-decrease rate control is a bad match for video. Comparing to IPSec, the proposed application-layer approach has less complexity and transmission overhead. Furthermore, performing authentication at the application layer makes it possible to design and adapt the authentication for each specific media object in order to optimize the authenticated media quality, given the available rate and network loss conditions. This is in contrast to SSL/TLS and IPSec approaches which treat the media as “data” and which do not explicitly consider media quality. A variety of examples are given throughout this paper to motivate and illustrate the benefits of media-aware authentication.

In this paper, after investigating existing end-end media authentication schemes (stream-based and content-based methods), we examine how to design authentication schemes for streaming media which are tolerant to packet loss and exploit the unequal importance of different media packets. Instead of optimizing the conventional authentication metric of packet verification probability, we optimize the quality of the authenticated media, which is determined by the packets which are received, decodable, and authenticatable. The quality of the authenticated media is optimized by allocating authentication resources unequally across streamed packets based on their relative importance, thereby providing unequal authenticity protection (UAP). Simulation results are then given using different media (image and video), different coding standards (JPEG, JPEG2000, and H.264), and different channels (wired and wireless) to demonstrate the improved performance which can be achieved.

The rest of this paper is organized as follows. In Section II, we define important terms and performance metrics used in this paper and review content-based and stream-based media authentication schemes. In Section III, we describe our approach for authentication of streaming media using UAP and how we apply it to different media, different compression standards, and for different channels, and provide experimental results which illustrate its performance. Additional issues are discussed in Section IV, and we conclude with a summary in Section V.

II. Preliminaries

In this Section, we first introduce some concepts and terms related to media stream authentication. We then give a detailed review on existing content-based and stream-based authentication approaches, and then describe some important metrics for assessing authentication performance for media streaming.

A. Concepts and definitions

Authentication, integrity and non-repudiation: Usually authentication is associated with data integrity, source identification, and non-repudiation, because these issues are very often related to each other: Data which has been altered effectively should have a new source; and if the source cannot be determined, then the question of alteration cannot be settled either. Typical methods for providing data authentication are digital signature schemes (DSS) and message authentication codes (MAC). Digital signatures use an asymmetric (public/private) key pair, while MACs use a symmetric (private) key. Both DSS and MAC techniques build upon the use of one-way hash functions.

One-way hash function: A one-way hash function or cryptographic hash is a hash function that works only in one direction to generate a fixed-length bit-string for any given data with arbitrary size. These hash functions guarantee that even a one-bit change in the data will result in a totally different hash value. Therefore, the use of a hash function provides a convenient technique to identify if the data has changed. Typical hash functions include MD5 (128bits) and SHA-1 (160bits).

Message authentication code: To prevent an attacker from both changing the data and replacing the original hash value with a new one associated with the new data, keyed hash functions are used where the hash is computed of a combination of the original data and a secret key. Keyed hashes correspond to one of the most important types of MACs.

Digital signature schemes: These schemes include (1) a procedure for computing the digital signature at the sender using the sender's private key, and (2) a procedure for verification of the signature at the receiver using the associated public key. Computing a digital signature is very computationally expensive, and depends on the length of the data being signed. Therefore, instead of directly signing the data, the typical approach is to compute a hash of the data and then sign the hash value. Public key DSS is a common technology and has been adopted as an international standard for data authentication [11], where the private key is used for signature generation and the public key is used for signature verification. The generated signature is usually about 1024bits.

Media data versus media content: Given a specific type of multimedia (e.g., image), the term media "data" refers to its exact representation (e.g., binary bitstream) while the term media "content" refers to the semantics of the same data representation. The term *semantics* refers to the aspects of meaning that are expressed in a language, code, or other form of media representation. For example, after lossy compression the original and reconstructed media data is different, however the media content or media semantics should be the same (e.g., the same people are visible in both the original and reconstructed image). Semantics measurement is generally subjective, and is a function of the specific applications. For example, matching or similarity score is the most common one used in pattern recognition.

Content authentication: The term "content authentication" refers to verifying that the meaning of the media (the " content" or semantics) has not changed, in contrast to data authentication which considers whether the data has not changed. This notion is useful because the meaning of the media is based on its content instead of its exact data representation. This form of authentication is motivated by applications where it is acceptable to manipulate the data without changing the meaning of the content. Lossy compression is an example.

Incidental distortion and Intentional distortion: Incidental distortion refers to the distortion introduced from coding and communication like compression, transcoding, and packet loss, etc. Intentional distortion refers to the distortion introduced by malicious attacks like image copy-paste (e.g., changing the text in a picture), packet insertion, etc. In some applications, the goal of the authentication scheme is to tolerate incidental distortions (i.e., all impacted media caused by incidental distortions will still be deemed as authentic media) while rejecting or identifying intentional distortions.

This paper focuses on media data authentication, the above media content authentication discussion is provided to give the reader a better understanding of the broader research field.

B. *Performance metrics for streaming media authentication*

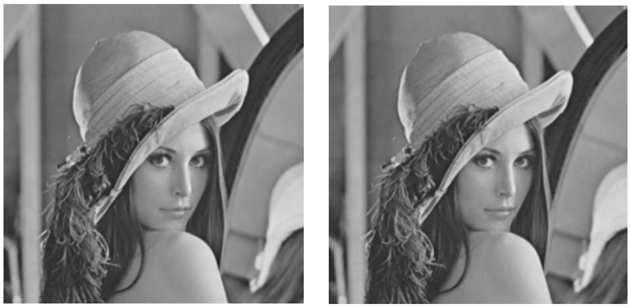
Verification probability: Probability that a received packet is also verifiable (authenticatable). Ideally, all received packets can be verified, however this leads to high overhead and computational costs, motivating the need for alternative techniques which also provide high verification probability but at significantly lower costs.

Computation overhead: The computational resources required to generate the signature at the sender and to verify the signature at the receiver. As the media stream typically involves a huge amount of continuous data, this requirement becomes even more critical when the receiver is a mobile device with limited computational capabilities.

Communication overhead: The additional rate associated with the authentication information which is transmitted along with the media content. The additional rate may include MAC values, digital signatures, or hashes. It is important to minimize this overhead, especially in settings where the total rate available is limited, since it directly reduces the rate available for source or channel coding.

Sender delay: The additional delay placed on a packet before it can be transmitted because of the authentication processing (e.g., processing a block of packets). In real-time communication scenarios, a high sender delay often requires a large buffer at the sender.

Receiver delay: The delay from the time when a packet is received to the time when can be authenticated by the receiver. A high receiver delay often requires a large buffer at the receiver. For streaming media, usually each packet has an associated playout deadline after which it becomes useless, therefore the receiver delay from authentication should be designed so that the packet does not miss its deadline.



(a) JPEG coded with quality = 10. (b) JPEG coded with quality = 4.

Fig.2. Example of two versions of an image which contain similar semantic content but different coded data.

C. *Content-based authentication*

In this subsection we describe content-based authentication techniques [18][19][20][21][22] which form one class of possible solutions to achieve end-to-end media stream authentication. The goal of this class of approaches is to authenticate the media at the content level, and the rationale to support this type of solution is depicted in Fig. 2 for the gray-scale image *Lena* (512x512 pixels). The image is coded on the left with JPEG at a quality level 10 (best quality) and on the right with JPEG quality 4 (good quality). The file size is reduced from 151 Kbytes for the best quality to 36 Kbytes for good quality. While the two bitwise representations of the image are completely different, most of the semantic meaning of the *Lena* image is still preserved on the right, which implies that media content is mainly comprised of perceptually "invariant" features. This motivates the idea that authenticating media content can be achieved by authenticating these invariant features, as opposed to the media data.

TABLE I

CONTENT-BASED AUTHENTICATION (SIGNING)

```
System setup
▪ Content owner requests a pair of keys (private key and public key)
  from the PKI authority.
▪ Select an adequate ECC scheme ($N, K, D$) given domain-specific
  acceptable manipulations. Here $N$ is the length of output encoded
  message, $K$ is the length of original message and $D$ is the error
  correction capability.
▪ Select another ECC scheme ($N', K', D'$) for watermark formation
  as described above (optional)
Input
Original image to be signed $I_o$
Begin
▪ Partition image into non-overlapping blocks ($1..B$).
▪ For block 1 to block $B$, Do
      Conduct block-based transform such as DCT.
      Extract invariant features robust to those acceptable manipulations
      Map each feature set into one or more binary messages, each of
      which has length $K$.
      ECC encode each binary vector to obtain its codeword $W$ ($N$ bits)
      and parity ($N$-$K$ bits).
          i) Take all parity bits as the watermark.
             ECC encode it using the scheme ($N', K', D'$).
             Embed the watermark into the selected blocks;
             Inverse transform to obtain the watermarked image $I_w$;
          ii). Collect codewords from all blocks $W(1..B)$;
             Concatenate them to form a single bit sequence Z
End
Hash the concatenated codeword sequence $Z$ to obtain $H(Z)$;
Sign $H(Z)$ using the owner's private key to obtain the Signature $S$;
End
Output:
▪ Watermarked image $I_w$;
▪ Content-based crypto signature $S$.
```

The basic idea of signature generation or signing an image is described in Table I. Signature verification is usually the inverse process of signature generation for the standardized digital signature schemes [23], and is not depicted. The selected feature set, extracted from the selected blocks, is robust to a pre-defined set of acceptable manipulations (e.g., packet loss) while sensitive to malicious attacks (e.g., copy-paste). The incidental distortions on feature values caused by those pre-defined acceptable manipulations can be further eased by employing an error correction coding (ECC) scheme. Watermarking is employed here to hide the overhead of ECC (i.e., parity check bits). The digital signature is then generated using the content owner's private key to sign the hash value of all the concatenated ECC codewords. More application-oriented solutions under such framework have been proposed in [20][21][22] which are robust to different pre-defined acceptable manipulations like multi-cycle lossy compression by JPEG or JPEG2000, format conversion and packet loss, etc. In [22], to tackle the unpredictable quality degradation from packet loss, we further apply pre-processing and block shuffling to the image before signing to stabilize the feature extracted at the receiver end.

In addition to its robustness to the pre-defined acceptable manipulations and the perceptually good watermarked media quality, the key attribute of these content-based authentication schemes are their compatibility with the Public Key Infrastructure (PKI) [23] which is the most popular data authentication protocol in today's Internet. Also, the aforementioned scheme (as well as a number of other techniques such as [8]) have been adopted into the JPEG2000 Security (JPSEC) international standard [24].

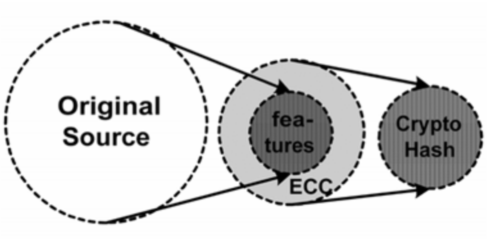


Fig.3. System security illustration on content-based authentication

System security plays a vital role in an authentication system. From our previous description, we can see that three modules mainly affect system security, i.e., feature extraction, ECC, and hashing (note the security of typical standardized digital signature schemes like DSA is usually high). Therefore, the security performance of the system may be measured in terms of the probability of the system being cracked, i.e., given an image, the probability of finding another image which can pass the signature verification, under the same parameters. To highlight some of the issues and tradeoffs involved, consider the simplified conceptual example where the system security can be expressed by three mutually independent probabilities corresponding to the probabilities that any of the 3 modules can be cracked: P_F for feature extraction module, P_E for ECC module, and P_C for hashing. For a secure system each of the probabilities should be very small, allowing us to approximate the overall system security as:

$$P = 1-(1-P_F)(1-P_E)(1-P_C) \approx P_F + P_E + P_C \quad (1)$$

Since P_C is much smaller[1] than P_F and P_E, we focus attention on P_F and P_E. In fact, P_F and P_E impair the system security in different ways, as shown in Fig. 3. A good feature descriptor should represent the original source as close as possible. In contrast to feature extraction which removes redundancy from the original source, ECC adds redundancy in order to tolerate incidental distortions. Hence a good feature set and a proper ECC scheme are key factors in system security.

The above simplified analysis highlights that content-based authentication schemes have to tolerate a certain False Acceptance Ratio (FAR) and False Rejection Ratio (FRR), in a similar manner to other pattern recognition systems (e.g., biometrics). FAR means a number of unauthentic or attacked content will be verified by the system as authentic, while FRR means a number of authentic content will be verified by the system as unauthentic. In other words, there are a number of media files whose authenticities cannot be accurately identified by the system. Obviously, the FAR and FRR of a system will effect its potential applications.

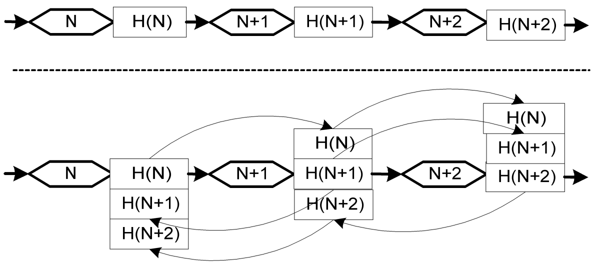


Fig.4. Example media stream authentication graphs. The bottom graph is resilient to the loss of a single packet, unlike the top graph.

In summary, content-based authentication schemes provide the ability to authenticate content which has undergone acceptable manipulations, as long as the content features are preserved. However, these schemes have limitations on the acceptable FAR and FRR, which limit their usefulness in many applications.

D. *Stream-based authentication*

The second class of possible solutions to achieve end-to-end media stream authentication is to directly authenticate at the stream or packet level. The system security can be mathematically proven as it is based on conventional data security approaches, though its system robustness is not as strong as content-based authentication. For example, it may only be robust to packet loss, and not to other manipulations.

Stream-based authentication can be further classified into error correction coding (ECC) based methods [25][26] and graph based methods [27][28][29][30][31][32][33][34][35][36][37][38]. [25] proposed to use erasure code (a type of ECC code) for stream authentication. For each block, the digital signature is coded with an erasure code and is then dispersed across the packets. As long as the number of lost packets is less than a threshold, all received packets can be authenticated. However, this scheme has high computational overhead due to the erasure coding. In addition, it also suffers from a high receiver delay, because the receiver has to wait for a minimum number of the received packets for authentication. [26] proposed a similar scheme, but with the additional goal of robustness to pollution attacks where adversaries inject false packets. This paper continues by focusing on graph-based authentication for media streaming.

[1] For a crypto hash function like SHA-1 (160 bits), P_c is about 10^{-79} under brute-force attack. Typical values for P_E and P_F are around 10^{-N} with N in the range of [3, 10] depending on the feature extraction and ECC methods.

The basic idea of graph-based authentication is illustrated in Fig. 4. In the simple case, for each packet its hash is computed and appended to the end of the packet, as shown in the upper part of Fig.4, and the signature is computed across all of the hashes and sent separately. Authentication is performed after receiving the last packet. However, this scheme fails when packet loss occurs because the signature was generated based on all the hash values from all the packets. To overcome this problem, a straightforward solution is to add redundancies (e.g., additional edges in the graph) by attaching several hashes from other packets into the current transmitting packet. If the current packet (e.g., *N*) is lost, its hash can still be obtained from other packets (e.g., *N*+*m*).

Quite a number of graph-based stream authentication schemes have been proposed. [27] proposed an authentication scheme using a simple hash chain. It has low overhead and low receiver delay, but it also has a high sender delay and cannot tolerate any packet loss. [28] proposed a scheme based on the Merkle authentication tree [29], and it has a very high communication overhead, although it can tolerate any number of packet losses. [30] proposed the Efficient Multi-channel Stream Signature (EMSS) scheme, which uses a hash chain where each packet contains the hashes of previous packets and the signing is on the last packet. This scheme has a high receiver delay and a low sender delay. [31] proposed an alternative called the Augmented Chain (AC), however since the signing is still on the last packet, it also has a high receiver delay. [32] proposed an authentication scheme based on the Expander Graph (EG). It has a very large communication overhead which is unacceptable for most applications. [33] proposed the Random Graph (RG) scheme, where the signing is on the first packet, and each packet contains the hashes of every subsequent packet with certain probability. Therefore it also has a high communication overhead. [33] also examined the problem of prioritizing packets through the use of different amounts of redundancy to achieve different verification probabilities. [34] proposed a butterfly graph for stream authentication, which aims to achieve low overheads and high authentication probability. The scheme is robust against both random and burst packet loss and outperforms the existing schemes in terms of overhead, verification probability, and receiver delay. Additional work on graph-based stream authentication adopting redundancy can be found in [35][36][37][38].

It is worth noting that some proposed stream authentication methods have proven their optimality in terms of verification probability, i.e., achieving the optimal verification probability given a fixed overhead and assumed loss model [31]. However, we believe that authenticating media streams still demands improved solutions because of the following two intuitive considerations.

First, previous approaches assume and treat all packets as if they are of equal importance, which generally is not true for media packets. For example, packets containing P-frame[2] coded video data are typically more important than those containing B-frame coded video data. To illustrate the distribution of packets' importance, we use JPEG-2000 to encode the "bike" image (one of the JPEG-2000 standard test images) with 16 layers and 80 JPEG-2000 packets[3] per layer, and compute the distortion reduction for every individual packet, which is depicted in Fig. 5. The amount of distortion reduction per packet exhibits huge differences. Out of the 2560 packets, 2464 packets (more than 96%) have a distortion reduction less than 100 Mean Square Error (MSE) units, and the other 96 packets (less than 4%) have much greater distortion reduction. In other words, a small number of packets are much more important than the rest of the packets. Note that this characteristic is often exploited via unequal error protection to transport media data over lossy networks. Similarly, stream authentication can also utilize this characteristic by trading off authentication redundancy based on packet importance: increasing the redundancy degree for more important packets, so as to increase their verification probability [33], and reducing it for the less important packets which have a smaller effect on reconstructed visual quality. We believe that this approach can be more practically useful for media applications than conventional authentication approaches which do not account for the varying importance of each media packet.

Second, in contrast to generic data stream authentication where verification probability is deemed as the primary performance measure to be optimized, for media stream authentication the media quality of the authenticated media often is a more important metric. Therefore, we believe that media quality is a more important metric for optimization than verification probability.

The next section describes several media-oriented stream authentication schemes designed using a relatively new framework for rate-distortion-authentication (RDA) optimization., which accounts for the unequal importance of different packets and tries to optimize the authenticated media quality.

III. CONTENT-AWARE MEDIA STREAM AUTHENTICATION OPTIMIZED FOR QUALITY

This section begins by formulating a generic rate-distortion-authentication optimization framework for media streaming. We then present three examples which illustrate its application for different media (JPEG-2000 and JPEG for images, H.264 for video) delivered over different lossy channels. These examples are intended to convey the basic design principles, without distracting the reader by the specific details of each realization. Please see the cited references for specific details.

[2] Typical video compression standards such as MPEG code each frame as an intra (I) frame, predicted (P) frame, or bi-directionally predicted (B) frame.

[3] Note that a JPEG-2000 packet is distinct from a network packet, which typically holds multiple JPEG-2000 packets.

A. *Proposed RDA framework optimized for quality*

The problem of authenticating a media stream can be solved in an Rate-Distortion-Authentication Optimization framework, which constructs an authentication graph trading off two conflicting goals: minimizing total rate (coded media rate and overhead) and minimizing total expected distortion (or maximizing media quality) of the authenticated media. Given a specific rate and network condition, the goal is to compute an authentication graph that minimizes the total expected distortion of the authenticated media. Conversely, the optimized graph minimizes the overall rate, for a specific target distortion and given network condition. In other words, the rate-distortion performance of the optimized authentication graph lies on the lower convex hull of the set of all achievable rate-distortion performances. The following formulation builds on and was partially motivated by the rate-distortion optimization framework and notation proposed in [41] for solving the problem of packet scheduling for media streaming. We propose an enhanced framework which encompasses joint source coding rate, distortion and authentication optimization for media delivery. Note that the straightforward coupling of R-D streaming techniques (e.g., [41]) and stream authentication techniques does not yield a satisfactory solution. Instead a careful joint design of the two is necessary. For example, [41] proposes a solution given a coding dependency graph among the packets. When authentication is added the situation is more complicated as authentication not only introduces additional overhead but also a second dependency graph among the packets. Therefore, careful media-aware design of the authentication graph and joint rate-distortion-authentication optimization can provide significantly improved performance.

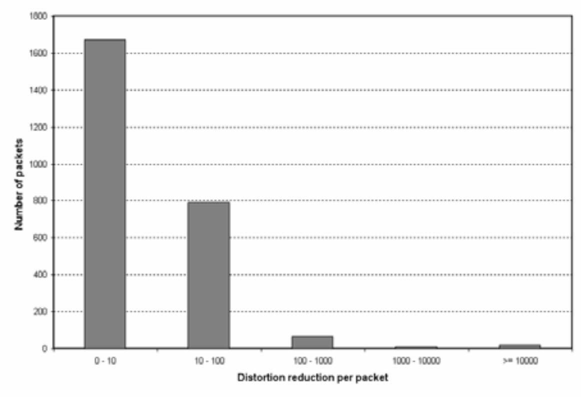


Fig.5. Distribution of packets' distortion reduction in a JPEG-2000 image

An authentication graph is a directed acyclic graph denoted by <*V, G*>, where *V* is the set of nodes and *G* is the set of directed edges in the graph. A node in *V* corresponds to a media packet or a signature packet signed with a crypto signature scheme, and there is typically only one signature packet in *V*. A directed edge $e(i,j)$ from node P_i to P_j indicates that the hash value of P_i is appended to P_j, where P_i and P_j are referred to as the source node (or source packet) and target node (or target packet), respectively. The edge $e(i, j)$ is also referred to as a hash link that connects P_i to P_j. The redundancy degree of the packet P_i is the number of edges coming out of P_i. In particular, the redundancy degree is 0 for a signature packet. At the receiver, the nodes corresponding to the lost packets are removed from the graph. A packet P_i is verifiable if there remains a path from P_i to the signature packet. The verification probability is the probability that a packet is verifiable given that it is received.

To formulate the rate-distortion-authentication optimization problem, we define the vector variable $\pi = [\pi_0, \pi_1, ..., \pi_m, ..., \pi_{M-1}]$, where π_m is the set of target nodes of the edges coming out of P_m. The redundancy degree of P_m is $|\pi_m|$, where $|\pi_m| \geq 1$. Given the set of nodes *V*, the variable π uniquely defines the authentication graph. Denoting the total rate (sum of source, channel, and authentication rates) as *R* and the overall expected distortion as *D*, our goal is to find the optimal π^* that minimizes the expected Lagrangian in (2) for a given $\lambda>0$. The Lagrange multiplier λ is used to control the trade-off between the rate *R* and the expected distortion *D*. For instance, a smaller value of λ will result in an optimized policy leading to smaller expected distortion *D* and higher overhead *R*, and vice versa.

$$\pi^* = \arg \min_{\pi}(D + \lambda R) \qquad (2)$$

with $D = D_s + D_c + D_a$ and $R = R_s + R_c + R_a$, and where we assume that the distortions and rates are additive. The source rate R_s and distortion D_s define the rate and distortion after compression. Similarly channel rate R_c and distortion D_c define the rate increase (from introducing redundancy) and distortion gain (i.e., recovered quality) [1][39]. The authentication rate R_a is the extra bytes introduced for media authentication, e.g., the rate for all of the hashes appended to the packets and the digital signature. Its rate $R_a(\pi)$ can be computed as in (3), where SIZ_{Sig} and SIZ_{Hash} are the sizes of the signature and hash respectively.

$$R_a(\pi) = SIZ_{Sig} + \sum_{P_m} |\pi_m| SIZ_{Hash} \qquad (3)$$

The expected authentication distortion $D_a(\pi)$ can be calculated as in (4), again assuming distortion is additive, where D_0 is the distortion when no packet is consumed because of authentication, ΔD_m is the amount by which the distortion will be reduced if packet P_m is consumed, ρ_m denotes the probability that P_m is decodable, and $1 - \varepsilon(\pi_m)$ denotes the probability that P_m is verifiable with π_m given P_m is decodable.

$$D_a(\pi) = D_0 - \sum_{P_m} \Delta D_m \rho_m [1 - \varepsilon(\pi_m)] \qquad (4)$$

Achieving the global optimization of π^* in (2) is generally computationally impractical, since one has to consider many factors from source coding, channel coding, and authentication and their couplings. A more practical approach to compute a solution to this problem is to begin by first considering overall resource allocation among source coding, channel coding, and authentication, followed by iteratively performing independent optimization across each of them. Depending on the specific application, one could further simplify (2) by omitting R and S components to make (2) analytically solvable and then employing some empirical approaches for directly assigning some parameter values. For instance, in the following application to scalable image coding scheme, instead of computing the authentication overhead for each packet, we simply categorize all packets into 3 classes and then fix the overhead for each class. The following subsections highlight how to realize the optimization together with experimental results on different media with different compression formats under different channel conditions.

B. *Application to scalable image coding scheme [39]*

In this subsection, we demonstrate how we can use information about the media content to achieve quality-optimized end-to-end stream authentication. For this purpose, we temporarily ignore source and channel factors (refer to eqn. (2)). We examine scalable media coding because it encodes the media in such a way that the resulting bitstream corresponds to different sets of bits of differing importance. Such concept actually is very close to our idea for media stream authentication. We choose the latest image coding standard JPEG-2000 [24] because of its great potential for navigating or streaming very large images such as maps, satellite images and motion images. Another reason is, during JPEG-2000 encoding, each so-called JPEG-2000 packet is associated with a quantity ΔD, which is the amount by which the overall distortion will be reduced if the packet is consumed by the decoder. A natural and intuitive idea for exploiting information about the content for authentication is as follows: For more important packets (i.e., larger ΔD), to increase their verification probability we can replicate and append their hashes in greater numbers to other packets, which increases their verification probability (and also the overhead). Conversely, we can allow lower verification probability for the less important packets in order to lower the overhead.

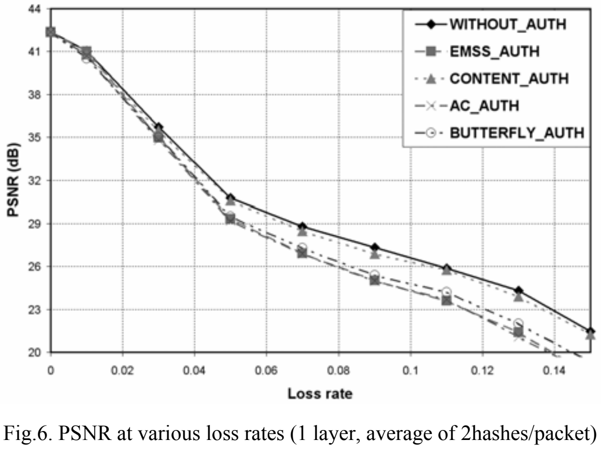


Fig.6. PSNR at various loss rates (1 layer, average of 2hashes/packet)

To demonstrate the effectiveness of adapting the authentication redundancy to the distortion, we encode the image using JPEG-2000 with only 1 layer, so the proposed solution CONTENT_AUTH can take advantage of the distortion information but not the layer structure. In CONTENT_AUTH, we empirically categorize all packets into three classes of equal number of packets according to their importance (i.e., if it is lost, how much distortion it will incur, refer to Fig. 5). For the most important packets, we repeat their hashes three times by appending them to other packets. The middle importance packets have their hashes repeated twice, while the least important have them repeated once. So the redundancy degree is 2 on average. For comparison, the alternative schemes of EMSS_AUTH [30], AC_AUTH [31] and BUTTERFLY_AUTH [34] are applied using a similar level of redundancy. To provide a benchmark for the achievable distortion, the scheme WITHOUT_AUTH is used where we simply send the packets in the order they appear in the JPEG-2000 code-stream, and no authentication is applied. This scheme provides a reference for the achievable distortion performance if verification is not required, and therefore also provides an upper bound on the performance of any authentication scheme. Fig. 6 plots the PSNR of the five schemes tested. CONTENT_AUTH consistently outperforms the other schemes at all network loss rates. In fact, the PSNR curve of CONTENT_AUTH is very close to that of WITHOUT_AUTH, providing further evidence for the benefit of applying content-aware authentication.

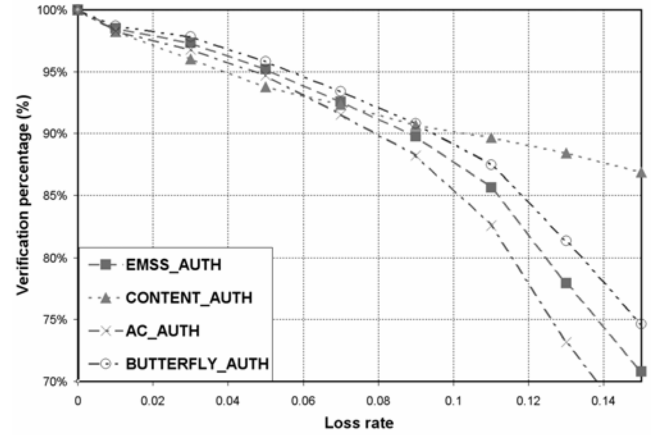
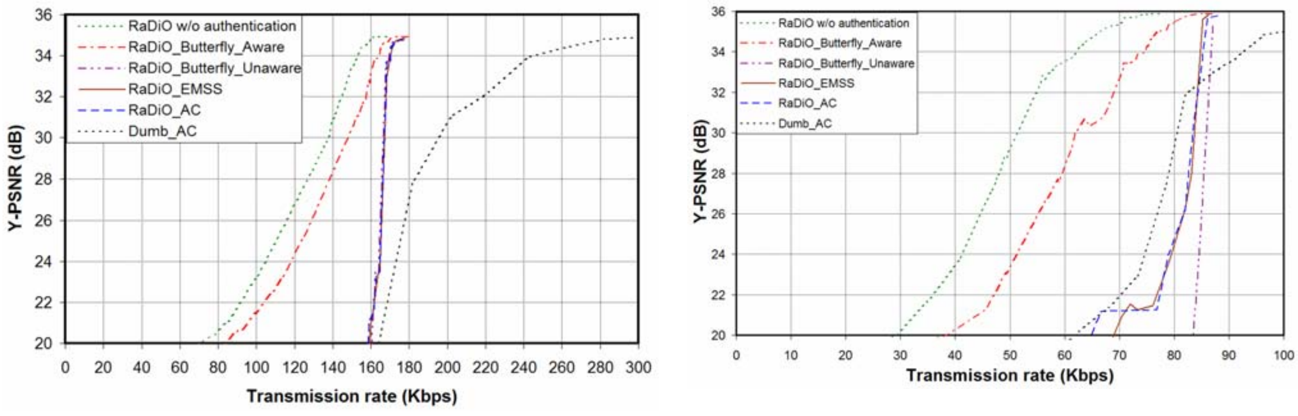


Fig.7. Verification probabilities at various loss rates (2 hashes / packet on average, with 1 layer)

Fig.7 shows the verification probabilities for the four authentication schemes. When the loss rate is less than 0.1, CONTENT_AUTH has a slightly lower verification probability, because one third of the packets have redundancy degree of 1. When the loss rate is larger than 0.1, a flat redundancy degree of 2 for all packets is not sufficient, which causes a dramatic decrease for EMSS_AUTH, AC_AUTH and BUTTERFLY_AUTH. For CONTENT_AUTH, the decrease is much smaller because one third of the packets have redundancy degree of 3. Figures 6 and 7 demonstrate that while CONTENT_AUTH sometimes has lower verification probability than the other authentication schemes,



(a) Foreman (QCIF)

(b) Container (QCIF)

Fig. 8 – R-D curves for the following systems when streaming over a network with 3% packet loss and time-varying delay: (1) conventional *RaDiO* without authentication, (2) *RaDiO_Butterfly_Aware*, (3) *RaDiO_Butterfly_Unaware*, (4) *RaDiO_EMSS,* (5) *RaDiO_AC,* and (6) *Dumb_AC*.

it still produces higher PSNR. Therefore, CONTENT_AUTH provides improved distortion-overhead performance because its authentication overhead is added in a more cost-effective manner --- it is guided by the content importance.

C. Application to non-scalable video coding scheme [40]

In this section, we describe authentication-aware R-D optimized streaming for authenticated video. Based on each packet's importance in terms of both video quality and authentication dependencies, the proposed technique computes a packet transmission schedule that minimizes the expected end-to-end distortion of the authenticated video at the receiver subject to a constraint on the average transmission rate. This work was motivated by recent advances in Rate-Distortion Optimized (RaDiO) [41] streaming techniques, which compute a packet transmission policy that minimizes the expected end-to-end distortion at the receiver subject to a constraint on the average transmission rate.

In the following we assume the case of pre-encoded video to be streamed, for example for video on demand services. Given a compressed video with associated authentication information, the first step is to compute the important quantities associated with each packet. The distortion increment, packet size, and display time are the same as in conventional *RaDiO* techniques [41]. The overhead size can be computed from the topology of the authentication graph. Second, at every transmission opportunity, the R-D optimization process selects the best packet(s) for transmission based on their parameters. For example, packets with higher importance (distortion increment + authentication importance) and smaller size (packet size + overhead size) are assigned more transmission opportunities. In summary, we formulate a rate-distortion-authentication optimization problem to minimize the expected distortion of the authenticated video at the receiver, subject to the constraint on average transmission rate (refer to (2)). Please recall that unlike conventional RaDiO where all packets received before their associated playout deadline contribute to improve the media quality, in our case only the received and authenticated packets contribute, i.e., a packet which is received but not authenticated is equivalent to being lost.

Further information about the algorithm is given in [40], here we highlight the algorithms performance via simulation results using the latest video compression standard H.264. In Figure 8 we plot the R-D performance with 3% packet loss and time-varying delay. RaDiO implements the original RaDiO without authentication, whose performance is used as the upper bound for all other systems. Dumb_AC implements a straightforward transmission of video packets protected with Augmented Chain which is claimed optimal for generic data streaming [31]. Authentication-aware RaDiO streaming, incorporating joint optimization of RaDiO and authentication, and using Butterfly dependency graph for authentication is examined in RaDiO_Butterfly_Aware. It is used to illustrate the performance achievable by an authentication-aware RaDiO technique. RaDiO_Butterfly_Unaware (i.e., no joint optimization between RaDiO and authentication) implements authentication-unaware RaDiO with Butterfly Authentication. It is the same as RaDiO_Butterfly_Aware except that it uses authentication-unaware RaDiO, and therefore the gap in performance between these two can be used to estimate the gain of "authentication awareness". RaDiO_EMSS and RaDiO_AC implement authentication-unaware RaDiO with EMSS and Augmented Chain respectively.

RaDiO_Butterfly_Aware outperforms all schemes, because it computes the transmission policy based on both packets' distortion increments and authentication importance. At low bandwidths, the authentication-unaware RaDiO fails as its R-D curve drops quickly to unacceptable levels. Nevertheless, at the same low bandwidth the proposed authentication-aware RaDiO provides an R-D curve which drops gracefully in parallel with the upper bound, given by RaDiO for unauthenticated video. However, we still notice that there is a performance gap between RaDiO and RaDiO_Butterfly _Aware (which is larger than the 8kb/s rate for authentication overhead) which remains as our future work.

As a further observation to understand the plots, from the sender's point of view, the channel capacity is $(1-e)^2 R_C$, where e is the packet loss rate and R_C is the channel bandwidth,

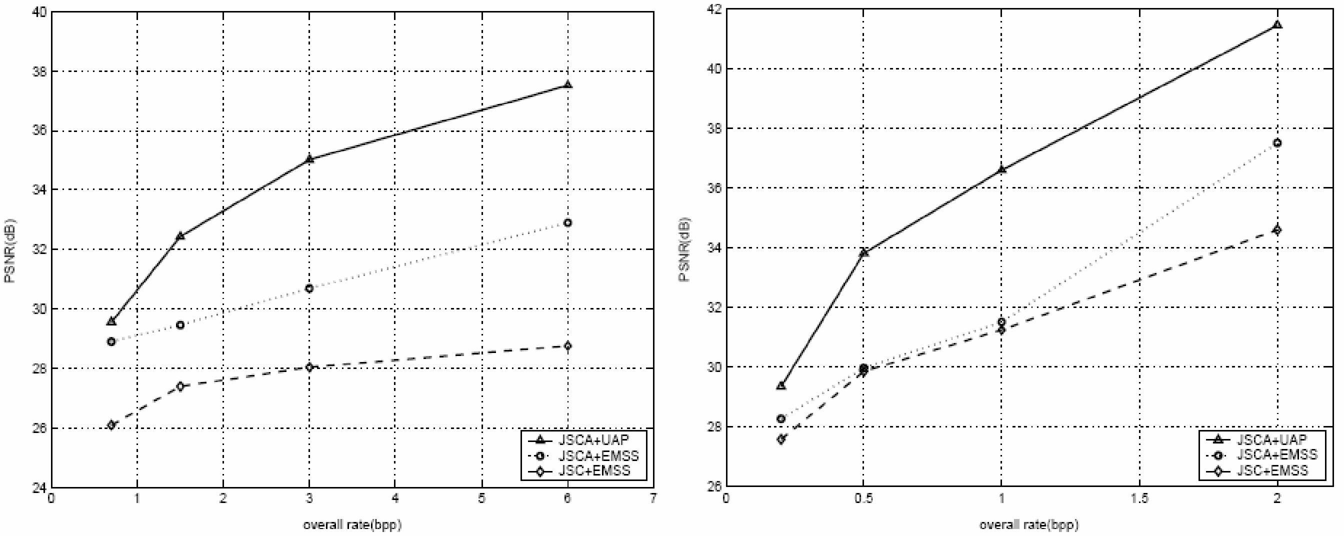


Fig.9 End-to-end R-D curves. Left: Lena at SER = 0.3. Right: Lena at SER = 0.01.

because the sender considers a packet as successfully delivered only after the packet is acknowledged by the receiver. Therefore, to transmit all packets at source rate R_S, the required bandwidth is $R_S/(1\text{-}e)^2$. More sophisticated acknowledgement schemes can reduce this required bandwidth to close to $R_S/(1\text{-}e)$ (depending on, e.g., constraints from the playout deadlines), however, we keep the current approach for conceptual simplicity. When channel bandwidth drops below $R_S/(1\text{-}e)^2$, the PSNR of authenticated video starts to drop, which is validated by all R-D curves provided. For example, in Fig.8(a), the source rate is 158kbps including 150kbps for video data and 8kbps for authentication overhead, therefore at a loss rate of 0.03 the knee of the R-D curve of RaDiO_Butterfly_Aware is located at $158/(1\text{-}0.03)^2$=168kbps.

Currently virtually all deployed video coding systems use non-scalable coding; however recent advances in scalable video coding may lead to its adoption in the near future. For additional details, as well as discussions of security services such as confidentially, authentication, and secure adaptation, see [42].

D. Joint source-channel-authentication scheme [43]

Lastly we describe an example of joint source-channel-authentication (JSCA). We can further derive the optimization function from (2) as follows, given a total bit rate budget:

$$D = \min_{R_s,R_c,R_a} \left(D_s(R_s) + D_c(\xi_a(e(R_c,R_a),\overline{m}(R_s,R_c,R_a)),e(R_c))\right) \quad (5)$$

where e again is the packet loss rate, $\overline{m}$ is the average hashes per packet, ξ_a is the optimal average weighted authentication probability over all packets (which can be obtained from a pre-designed look-up-table) [43]. This optimization can be achieved through searching the optimization parameters R_s, R_c, and R_a within the region of $0 \le R_s, R_c, R_a \le 1$ and $R_s + R_c + R_a \le 1$ in the (R_s, R_c, R_a) space. In our simulation, we only implemented a simple algorithm for finding the global optimal triplet (R_s, R_c, R_a) through exhaustive search.

To illustrate the potential benefits, we considered images coded with the JPEG standard using spectral selection progressive mode, because it was readily available, provided prioritization of the coded data, and facilitated estimating the source coding R-D curve by ρ-domain analysis [44]. For channel model, we assume a binary symmetric channel parameterized by symbol error rate (SER), which was then mapped to packet error rate. The proposed resource allocation scheme (JSCA+UAP) is benchmarked against two other schemes. In JSCA+EMSS the overall resource allocation is performed between source channel coding and authentication, but the resource within authentication is equally allocated across all packets using the basic EMSS scheme [30]. In the second scheme, JSC+EMSS, optimized resource allocation is performed across source and channel coding, however the overhead for authentication is fixed, and the basic EMSS [30] is again applied. Fig.9 shows that in each of the cases, JSCA+UAP which performs a joint optimization across source, channel, and authentication, provides as expected the best R-D curve, outperforming the other two schemes by around 3dB on average. Note that JSCA+EMSS also outperforms JSC+EMSS, especially when the channel

TABLE II
SOURCE / CHANNEL / AUTHENTICATION VS. SER (LENA, BPP RATE=2.5)

SER	R_s	R_c	R_a	PSNR (dB)
0.001	0.57	0.00	0.43	46.3473
0.01	0.55	0.08	0.37	44.7786
0.05	0.48	0.22	0.30	42.2841
0.1	0.36	0.36	0.28	39.7409
0.2	0.20	0.60	0.20	36.5461
0.3	0.12	0.78	0.20	33.7360
0.4	0.06	0.91	0.03	30.2107

distortion is severe.

To examine how the JSCA resource allocation is affected by channel conditions, we fix the total rate and examine how R_s, R_c and R_a vary, as the SER increases from 0.001 to 0.4. The results for Lena are illustrated in Table II, where we observe that when the channel condition is good, channel coding is unnecessary and most of the rate is allocated for source coding and authentication. As the channel condition degrades, a large portion of the total rate is allocated for channel coding. Also, as expected the PSNR of the authenticated image decreases as SER increases.

IV. Additional Comments

The prior sections described that the two classes of media stream authentication approaches (stream-based and content-based) can both be robust to packet loss. However, how to employ them for specific applications with different requirements is still a challenging issue. This section provides some additional comments on designing an application-oriented media delivery authentication system.

A clear understanding of the desired security service to be provided is critical. For example, what level of authentication security is required? What types of modifications to the media stream should be supported within that level of security? For example, what type of manipulations, and how many, should the system be able to authenticate. This can be thought of as the required robustness of the authentication. If the modifications are limited to packet erasures (as discussed in this paper) then the range of possible modifications for which the authentication should be robust to can be simply described --- greatly facilitating the analysis. In this case, important questions relate to the average packet loss rate, burst lengths, and general questions about what packet loss patterns may occur and how do they depend on the transmitted media stream (e.g., inter-packet spacing, packet lengths, etc). Other questions include, should the uncompressed media itself be signed so that the signed media stream can be authenticated across different coding formats? Should the signed media be robust to multiple lossy re-encodings or transcodings? Yes to any of these later questions may require authenticating the media at the content level, and the possible range of manipulations may be much larger and harder to describe, thereby making the security analysis much more difficult. Generally, as the number and range of acceptable content manipulations is increased, the provable achievable system security will decrease.

Stream-based and content-based authentication approaches provide complimentary benefits. The stream-based approaches are robust to packet loss (but typically not to other manipulations) while still maintaining the same system security as traditional data security techniques. Content-based approaches can be designed to be robust to a wide range of manipulations, however generally with lower mathematically provable or empirically tested levels of security. The above tradeoffs suggest combining the two classes of approaches, that is, jointly employing both stream-based and content-based methods to provide robustness to both packet erasures and other manipulations. This would involve a joint resource allocation across both stream-based and content-based authentication, and provides an interesting direction for future research.

V. Summary

In this paper we described how conventional data authentication techniques are not a good match for media streaming over a lossy packet network. When the coded media is loss tolerant, then it is beneficial for the authentication to also be loss tolerant. This paper reviewed existing end-to-end packet-loss-tolerant media authentication schemes including both stream-based and content-based methods. We then described how to design authentication schemes for multimedia streaming which are tolerant to packet loss and exploit the unequal importance of different packets. By applying conventional cryptographic hashes and digital signatures we can achieve a level of media security similar to that achievable in conventional data security. Instead of optimizing packet verification probability, we optimize the quality of the authenticated media, which is determined by the packets which are received, decodable, and authenticatable. The quality of the authenticated media is optimized by unequal authentication protection, which allocates authentication resources for each media packet according to its importance and coding dependencies. Performance improvements were illustrated using a number of simulation experiments with image and video coded using different compression standards. We believe that authentication for streaming media is an important technical problem, which will increase in practical importance as media streaming continues to gain in popularity.

Acknowledgment

The authors would like to sincerely thank Zhishou Zhang (I^2R), Zhi Li (I^2R) and Susie Wee (HP Labs) for their valuable discussions and contributions from system design, realization, to evaluation on various content-aware media streaming authentication schemes. The experimental results presented in this paper are merely for illustrative purposes, and more detailed and rigorous test results are given in [20][21][22][39][40][43].

References

[1] G. Sullivan and T. Wiegand, "Video Compression---From Concepts to the H.264/AVC Standard", Proceedings of the IEEE, pp.18-31, 2005.

[2] L. Kontothanassis, R. Sitaraman, J. Wein, D. Hong, R. Kleinberg, B. Mancuso, D. Shaw, and D. Stodolsky. "A transport layer for live streaming in a content delivery network", Proceedings of the IEEE, pp. 1408-1419, 2004.

[3] J. Li, "PeerStreaming: A practical receiver-driven Peer-to-Peer media streaming system," Microsoft Technical Report, MSR-TR-2004-101, Sept. 2004.

[4] E. Setton and B. Girod, "Rate-Distortion Analysis and Streaming of SP and SI Frames," IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No. 6, pp.733–743, 2006.

[5] A. Ali, A. Mathur and H. Zhang, “Measurement of Commercial Peer-to-Peer Live Video Streaming,” Workshop in Recent Advances in Peer-to-Peer Streaming, August, 2006.

[6] J. Apostolopoulos, “Secure media streaming & secure adaptation for non-scalable video,” IEEE International Conference on Image Processing, Oct 2004.

[7] S.J. Wee and J.G. Apostolopoulos, “Secure scalable streaming enabling transcoding without decryption”, IEEE International Conference on Image Processing, Oct. 2001.

[8] S. Wee and J. Apostolopoulos, “Secure scalable streaming and secure transcoding with JPEG-2000,” in Proc. IEEE International Conference on Image Processing (ICIP), 2003.

[9] S. Wee and J. Apostolopoulos, “Secure transcoding with JPSEC confidentiality and authentication,” in Proc. IEEE International Conference on Image Processing (ICIP), 2004.

[10] S. Imaizumi, O. Watanabe, M. Fujiyoshi and H. Kiya, “Generalized hierarchical encryption of JPEG 2000 codestreams for access control”, in Proc. IEEE International Conference on Image Processing (ICIP), 2005.

[11] B. B. Zhu, C. Yuan, Y. Wang, S. Li, “Scalable protection for MPEG-4 fine granularity scalability”, IEEE Transactions on Multimedia Vol. 7, No. 2, pp. 222-233, 2005.

[12] P. Moulin and R. Koetter, “Data-Hiding Codes”, Proceedings of the IEEE, Vol. 93, No. 12, pp. 2083--2127, 2005.

[13] C. Fei, D. Kundur and R. Kwong, “Analysis and Design of Secure Watermark-based Authentication Systems”, IEEE Transactions on Information Forensics and Security, Vol.1, No.1, pp.43-55, 2006.

[14] Bartolini, A. Tefas, M. Barni, and I. Pitas, “Image authentication techniques for surveillance applications”, Proceedings of the IEEE, Vol.89, No.10, pp.1403-1418, 2001.

[15] G. J. Simmons, *Contemporary Cryptography*, IEEE Press, 1992.

[16] The Transport Layer Security (TLS) Protocol Version. http://tools.ietf.org/html/rfc4346

[17] Security Architecture for the Internet Protocol: IPSec, http://www.ietf.org/rfc/rfc2401.txt

[18] Q. Sun and S.-F. Chang, “Signature-based media authentication,” in *Multimedia Security Handbook*, Chapter 21, pp.619-662, Edited by Furht and Kirovski, CRC Press, 2005

[19] C.-Y. Lin and S.-F. Chang, “A robust image authentication method distinguishing JPEG compression from malicious manipulation,” IEEE Transactions on Circuits and Systems for Video Technology, Vol.11, No.2, pp.153-168, Feb. 2001

[20] Q. Sun and S.-F. Chang, “A secure and robust digital signature scheme for JPEG2000 image authentication”, IEEE Transactions on Multimedia, 7(3), pp. 480-494, June 2005.

[21] Q. Sun, D. He and Q. Tian, “A secure and robust authentication scheme for video transcoding,” to appear in IEEE Transactions on Circuits and Systems for Video Technology, Vol.16, No.10, 2006.

[22] Q. Sun, S. Ye, C-Y. Lin and S-F. Chang, “A crypto signature scheme for image authentication over wireless channel,” International Journal of Image and Graphics, Vol. 5, No.1, pp 1-14, 2005.

[23] [ITU-T Recommendation X.509] (2005): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, 08/05. http://www.itu.int/rec/T-REC-X.509-200508-I

[24] Information technology - JPEG2000 image coding system: Security, ISO/IEC International Standard 15444-8, FDIS, 2006

[25] J. M. Park, E. K. P. Chong, and H. J. Siegel, "Efficient Multicast Stream Authentication using Erasure Codes," ACM Trans. Inf. Syst. Secur., vol. 6, no. 2, 2003

[26] C. Karlof, Y. Li and N. Sastry, “Authenticated block streams using error detecting erasure codes”, Technical report at http://www.cs.berkeley.edu/~nks/edec/bcast-class.pdf

[27] R. Gennaro and P. Rohatgi. “How to sign digital streams,” in Advances in Cryptology - CRYPTO '97, pp. 180–197, 1997

[28] C. K. Wong and S. Lam, "Digital Signatures for Flows and Multicasts", The University of Texas at Austin, Department of Computer Sciences, Technical Report TR-98-15. July 1998

[29] R. C. Merkel, “A certified digital signature,” in Advances in Cryptology—CRYPTO’89, LNCS 435, pp.218-238, 1990.

[30] A. Perrig, R. Canetti, J. Tygar and D. Song. “Efficient authentication and signing of multicast streams over lossy channels,” in Proc. of IEEE Symposium on Security and Privacy, pp. 56-73, 2000.

[31] P. Golle and N. Modadugu. “Authenticating streamed data in the presence of random packet loss,” ISOC Network and Distributed System Security Symposium, pp 13—22, 2001.

[32] D. Song, D. Zuckerman, and J. D. Tygar, “Expander Graphs for Digital Stream Authentication and Robust Overlay Networks”, In Proc. of the IEEE Symposium on Research in Security and Privacy, pages 258-270, May 2002

[33] S. Miner and J. Staddon, “Graph-based Authentication of Digital Streams”, In Proc. of the IEEE Symposium on Research in Security and Privacy, pages 232--246, 2001.

[34] Z. Zhang, Q. Sun and W-C. Wong, “A proposal of butterfly-graph based stream authentication over lossy networks,” in Proc. of IEEE International Conference on Multimedia & Expo (ICME), 2005.

[35] A. Pannetrat and R. Molva, “Efficient multicast packet authentication”, Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS’03), 2003

[36] T. Cucinotta, G. Cecchetti and G. Ferraro, “Adopting redundancy techniques for multicast stream authentication”, Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS’03), pp.183-189, 2003.

[37] C-F. Chan, “A Graph-Theoretical Analysis of multicast authentication”, Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03), pp. 155-160, 2003.

[38] Lysyanskaya, R. Tammasia and N. Triandopoulos, “Multicast Authentication in Fully Adversarial Networks, Proceedings of the 2004 IEEE Symposium on Security and Privacy, pp.241-248, 2004.

[39] Z. Zhishou, Q. Sun, W. Wong, J. Apostolopoulos, and S. Wee, “A content-aware stream authentication scheme optimized for distortion and overhead,” in Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2006. Complete version appears in IEEE Transactions on Multimedia, Vol. 9, No. 2, pp 320-331, 2006.

[40] Z. Zhishou, Q. Sun, W. Wong, J. Apostolopoulos, and S. Wee, “Ratedistortion optimized streaming of authenticated video,” in Proc. of IEEE International Conference on Image Processing (ICIP), 2006. Complete version is to appear in IEEE Transactions on Circuits and Systems on Video Technology 2007.

[41] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” IEEE Trans. on Multimedia, Vol. 8, No. 2, pp. 390 – 404, April 2006

[42] J. Apostolopoulos, “Architectural Principles for Secure Streaming & Secure Adaptation in the Developing Scalable Video Coding (SVC) Standard”, IEEE ICIP, 2006.

[43] Z. Li, Q. Sun, Y. Lian and C.W. Chen, “Joint source-channel-authentication resource allocation and unequal authenticity protection for multimedia over wireless networks,” in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), 2006. Complete version is to appear in IEEE Transactions on Multimedia, June 2007.

[44] Z. He, J. Cai, and C.W. Chen. Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding. IEEE Transactions on Circuits and Systems for Video Technology, 12(6), pp.511-523, June 2002.

First A. Author (M’xx-SM’xx-F’xx) and the other authors may include biographies at the end of regular papers. Biographies are not included in conference-related papers. This author became a Member (M) of IEEE in xxxx, a Senior Member (SM) in xxxx, and a Fellow (F) in xxxx. Other usual biography information includes birth date and place, education, employments, and memberships in other professional societies.