A SECURE AND ROBUST APPROACH TO SCALABLE VIDEO AUTHENTICATION

Qibin Sun, Dajun He, Zhishou Zhang and Qi Tian

Institute for Infocomm Research (I^{2}R),
21 Heng Mui Keng Terrace, Singapore 119613
Email: {qibin, djhe, zszhang, tian}@i2r.a-star.edu.sg

ABSTRACT

In this paper, we present a secure and robust content authentication scheme for scalable video streaming. In our authentication scheme we consider three common video transcoding methods as acceptable content manipulations, when the streaming bit-rate needs to be reduced, namely frame resizing, frame dropping and multi-cycle coding. By employing error correction coding (ECC) in different ways, the proposed scheme is insensitive to those incidental distortions introduced during the transcoding (i.e., robust) while is still sensitive to other intentional distortions such as frame alterations and insertion (i.e., secure). One key feature in our scheme is that it achieves an end-to-end authentication independent of transcoding infrastructure and obtains a good compromise between system robustness and security.

Keywords: Scalable video authentication, watermarking, digital signature, error correction coding.

1. INTRODUCTION

Consider the scenario of a station streaming video over the networks. It is important for the audiences to have guarantees that the video stream they are watching is indeed from the station. It is equally important to the station that only the content it sent be attributed to it. Such scheme could prevent the malicious parties from injecting commercials or offensive materials into the video stream.

Above problem has been well studied by the researchers in information security called streaming signing [1][2]. It is actually an extension from message signing by digital signature schemes which are able to both protect the integrity of the message and prevent the signer's repudiation. To adapt to the application of stream signing, people improved the digital signature schemes mainly on the following two aspects: One is the authentication efficiency for permitting authentication on the fly without introducing delays. The other is the authentication robustness for tolerating random loss of packets during streaming. However, their work still cannot meet our robustness requirements for authenticating scalable video streaming, where it usually adopts very flexible content-based transcoders such as frame resizing and dropping on the fly for dynamically adapting to the usage and condition changes in channels as well as clients [3].

Considering most popular transcoding approaches are frame resizing, frame dropping, quantization step size change, in this paper we propose a secure and robust authentication scheme for scalable video streaming, by employing ECC in different ways. Moreover, our solution achieves an end-to-end authentication independent of specific streaming infrastructure. The proposed scheme is extended from [4] where we proposed a new semi-fragile authentication framework for images in terms of ECC and Public Key Infrastructure (PKI).

The paper is outlined as follows: the related concepts and prior work are briefly introduced in Section 2. Section 3 describes our proposed scheme for scalable video streaming which is robust to frame resizing, frame dropping and re-encoding. Conclusions and future work are given in Section 4.

2. SECURE STREAMING AUTHENTICATION

Let's start from a simple case. Assuming the streaming is not required to dynamically adapt to channel and client conditions, then the simple way is to pre-encode the video content into several versions in terms of the compression bit-rates. The specific versions of the video content could then be streamed based on the requests from clients. For example, some clients may want the version with 512Kbits/s while others may only want the version with 32Kbits/s. In this case, video authentication also becomes simple: We can directly employ a typical digital signature scheme such as DSA [5], as illustrated in Figure 1. The station uses its private key to sign on the crypto hashes of different versions of the compressed video and generates their corresponding signatures. Such signatures are also sent to clients together with the videos in order to prove the authenticity of the video at the client site by using the station's public key. Considering that the stream may need to be verified part by part, a group of signatures could be obtained by partitioning packets into different groups and then signing on them to obtain a set of signatures.

0-7803-7965-9/03/$17.00 ©2003 IEEE

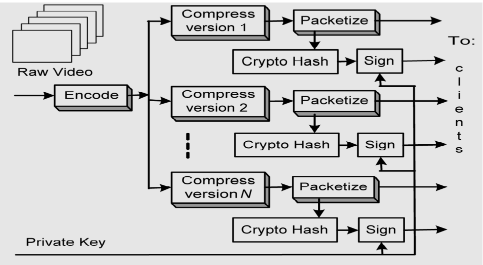

Figure 1. A simple solution for signing pre-coded streams

As signature signing is much more time-consuming than signature verification, a practical video signing system only signs on the last group of packets instead of signing packets group by group, as illustrated in Figure 2. The crypto hash of every group is XORed with the hashes of its previous groups. Then the station's private key will sign on the hash of the last group to form the signature of this video stream. At the client site, the client keeps repeating the same operation as at the station site. The whole video can then be verified after the client receives the signature and the last packet by using the station's public key.

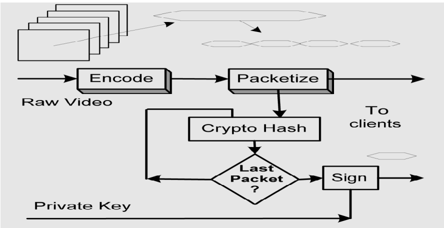

Figure 2. A practical solution for stream signing

When the video is streamed over unreliable channels such as User Datagram Protocol (UDP), some packets may be lost during the streaming. To overcome this problem, various approaches based on the concept of ECC [1, 2] are proposed. Refer to Figure 3 (Lower part), the basic idea is to add some redundancies by attaching several hashes from other packet groups into the current transmitting group of packets. If the current packet (e.g., *N*) is lost, its hash still can be recovered from other groups of packets (e.g., *N*+*m*). The verification on the whole video still can be executed. Obviously such solutions will result in extra transmission cost. Typically the authentication could tolerate 3% packet loss with the pay of 10% extra transmission load for hashes of other packets [2]. Hence it will become infeasible with the increase of authentication robustness.

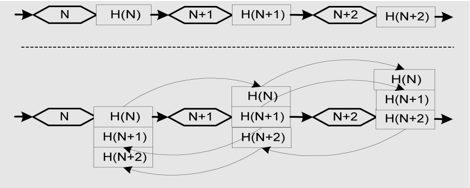

Figure 3. Stream authentication resilient to packet loss

The authentication issues we address in this paper is much more robust than previous work. In our application scenarios, authentication at packet or data level will not be able to meet the requirements in system robustness as well as the flexibility. Refer to Figure 4, there are several ways of transcoding video which adapts to channel and client conditions. In terms of the reduction of the streaming bit-rate, the video could be transcoded by means of either re-quantization, frame dropping or frame resizing. The transcoding can also be done flexibly either at the station site or at some intermediate routes. In terms of packet dropping, in some case, the packets may need to be dropped up to about 50% comparing to the amount of original packets. Therefore a new content-based solution is required for scalable video stream authentication.

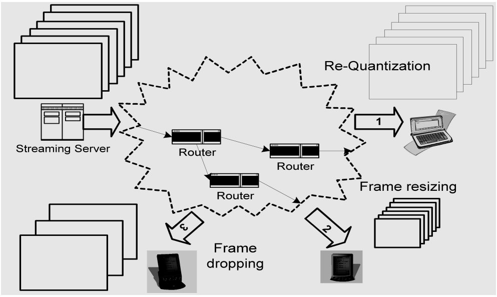

Figure 4. Typical transcoding for adaptive video streaming

3. CONTENT-BASED SCALABLE VIDEO AUTHENTICATION

3.1. System description

Considering the complexity and flexibility in streaming system design, we propose a content-based authentication which is independent of system infrastructure and achieves an end-to-end authentication, no matter where and how the video is transcoded by defined methods, the transcoded video should be authentic as long as its bit-rate is still within the pre-defined authentication strength (usually it is also in terms of bit-rate).

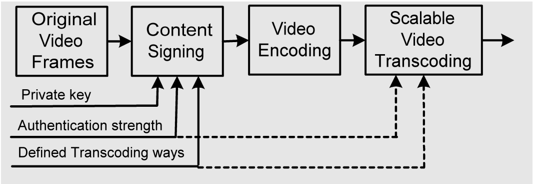

Figure 5. Content signing diagram

The brief diagram of our proposed scheme is shown in Figure 5. The signing works are based on frames (DCT-domain) in order to be independent of video coding and various transcoding approaches. Three inputs for content signing are: the station's private key, the authentication strength which means protecting the content to what degree (i.e., the video will not be deemed as authentic if it is trans-coded beyond this degree), and possible transcoding ways such as frame dropping, resizing, re-quantization or a combination among them. The signed raw video is then encoded and transcoded for streaming.

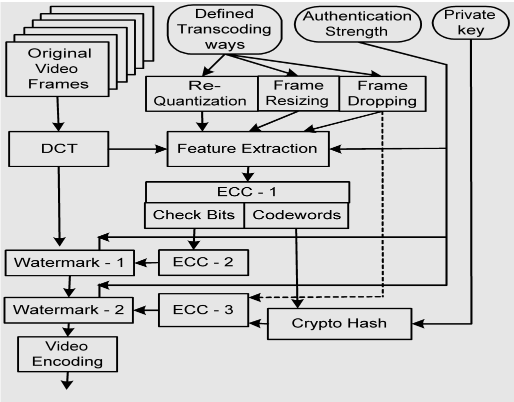

Figure 6. Detailed video stream signing

A more detailed illustration is shown in Figure 6. The whole scheme is extended from [4] where we adopted ECC and watermarking for generating a semi-fragile digital signature for JPEG images. Firstly the robust features are extracted from the content and then coded by an ECC scheme whose codeword can be separated into two parts: original message (in our case, it is extracted feature) and check information. The part of check information is embedded back into the video as a watermark. All ECC codewords are concatenated, then cryptographically hashed, and finally signed by content sender's private key to generate a content-based digital signature. The signature verification is almost an inverse procedure of content signing. A set of features as well as the embedded watermark is extracted from the video to be verified. Recall that the extracted watermark is actually the check information of ECC codewords at signing site. The video should be verified as authentic if it is not attacked, as the syndromes of new formed ECC codewords (The new extracted features as message part and the extracted watermarks as check information) are within correctable range.

Refer to Figure 6, as some transcodings are usually performed in DCT domain, we extract the content-based features and embed watermarks in DCT domain. After the features are extracted and ECC coded (ECC-1), their check information is coded by another ECC scheme (ECC-2) as watermark 1. The whole ECC-1 codewords are cryptographically hashed (e.g., SHA-1 or MD5) and part of hashes is ECC-3 coded as watermark 2. The final hash is signed by the station's private key to form the signature of this video.

As different acceptable transcoding approaches affect the robustness of extracted features and watermarking in different ways, we shall discuss them respectively. Note that if the video transcoding is a combination of the above-mentioned three defined approaches, the selected features should be an intersection of each extracted feature set.

- Re-quantization and multi-cycle coding

The first acceptable transcoding we consider is requantization as increasing quantization step size means a bit-rate reduction. Possibly multi-cycle coding will be involved. In such case the video authentication is required to be robust to new quantization and multi-cycle coding.

In [6], the authors proposed a content-based feature extraction and watermarking for image authentication, which is robust to JPEG compression. Two quantization step sizes are defined in the solution: Q_a is for generating features and watermarking while Q_c is for actual JPEG compression. They proved that as long as Q_c is less than than Q_a, the robustness of generated features as well as embedded watermarks can be theoretically kept. In [7], we incorporated their approach into PKI and improved its robustness to multi-cycle JPEG lossy compression: the message bits (features) are actually the quantized DCT values and check bits are their remainders. Therefore we can directly apply our solution [7] for this type of video transcoding.

- Frame resizing

The second acceptable transcoding we consider is frame resizing, i.e., change the frame size down to a smaller one. For instance, the conversion of the video from the Common Intermediate Format (CIF) to the Quarter Common Intermediate Format (QCIF) corresponds to a frame size with 352x288 down to 176x144. It requires our feature extraction as well as watermarking to survive under such frame resizing and preferably in the DCT domain for the purpose of computational reduction [8].

To meet this robustness requirement, we could perform feature extraction in QCIF instead of CIF. However, we cannot directly apply watermarking in such way as the watermarked video before streaming should still be with the original frame size (e.g., CIF) while the received video could be either in CIF or QCIF. In [9], we proposed a compressed-domain watermarking solution surviving CIF-to-QCIF conversion: watermark embedding is in CIF while the extraction could be either in CIF or QCIF.

- Frame dropping

The last acceptable transcoding we consider for bit-rate reduction is frame dropping. For instance, an original video with 25 frames per second can be transcoded to a new one with 5 frames per second by dropping 20 frames. Such transcoding also makes the authentication difficult as usually the frames are dropped before reaching the clients and no prior knowledge can be used for video verification such as which frames are dropped.

We solved this problem based on ECC, as illustrated in Figure 3. Its main idea is to not only embed into current frame (e.g., Frame *N*) with its ECC check information and crypto hash (e.g., Frame *N*-1) but also embed those hashes from other frames (e.g., Frame *N*-1, *N*-2, …N-*m*) [1, 2]. If some frames between N-1 and N-m are dropped, we can still obtain their corresponding crypto hashes from frame *N* for verification. The cost is to either reduce the length of crypto hash of each frame or increase the watermarking capacity as we cannot omit any ECC check information.

Assuming the watermarking capacity for each frame is 800 bits, the maximum frame-dropping rate is 80% (i.e., drop 20 frames out of 25 frames), and the length of ECC check and crypto hash for one frame is 600 bits (432 bits for ECC check, 128 bits for MD5 crypto hash and remaining bit for other purposes such as side information), then the total length of hashes of 25 frames is 3200 bits (128 bits x 25) while the whole available room for embedding hashes is 1640 bits ((800-600+128) bits x 5 frames). Therefore we can embed about 50% of hashes of total 25 frames, which means that we can only take about 64 hash bits from one frame instead of 128 bits. Note that reducing crypto hash length may cause some security risks; however, it is still acceptable in real applications as a strong contextual property in video content will make attacking content more difficult than attacking data.

3.2. Watermarking and its capacity

One of the challenges is watermarking capacity as we need to embed much information into each video frame such as ECC check for current frame and the hashes from other frames. The watermarking algorithms we adopted are from [9, 10] where they proposed to embed the watermarks by modifying the energies of a group of DCT coefficients among intra or inter blocks. We have tested that for a typical video sequence, the average embedding capacity for one frame is about 800 bits, which meets our requirements.

3.3. System robustness and security

For semi-fragile authentication, the robustness and security are two key and contradictive requirements in system design. The more robust the system is, the more risk it faces. Therefore we need to balance well between them.

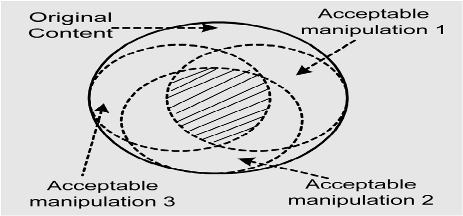
Original Content
Acceptable manipulation 1
Acceptable manipulation 3
Acceptable manipulation 2

Figure 7. The illustration of acceptable manipulations on content

The idea of robust feature extraction is shown in Figure 7. It basically requires the extracted feature be robust to defined manipulations. On the other hand, in terms of security it also requires that the extracted features should be able to represent the original video content as close as possible. Moreover, with more acceptable manipulations defined, the robust feature set should become smaller and smaller (the shaded area shows the feature space for three defined acceptable manipulations). Therefore three factors affect the system security: the first one is the method of feature extraction, the second one is ECC scheme selection and the last one is the number of defined acceptable manipulations. In [7], we have made similar but detailed analysis on system security.

4. SUMMARY

In this paper, we have proposed a robust and secure solution for video streaming authentication. We consider three common transcoding approaches as acceptable for our authentication solution. Some technical details are skipped due to the limit of paper size. Currently we are still improving our proposed solution and trying to extend for more practical applications such as on-line broadcasting. In such cases, we have to allow the new joining audiences to authenticate the stream.

5. REFERENCES

[1] R. Gennaro and P. Rohatgi, "How to sign digital stream", *Crypto'97*, pp. 180-197, 1997.

[2] J. M. Park, E. K. P. Chong and H. J. Siegel, "Efficient multicast packet authentication using signature amortization", *IEEE Symposium on Security and Privacy*, pp. 227-240, 2002.

[3] S. J. Wee and J. G. Apostolopoulos, "Secure scalable video streaming for wireless networks", *ICASSP'01*, 2001, USA.

[4] Q.-B. Sun, S.-F. Chang and K. Maeno, "A new semi-fragile image authentication framework combining ECC and PKI infrastructure", *ISCAS2002*, Phoenix, May, 2002.

[5] B. Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., 2nd edition, 1995.

[6] C.-Y. Lin and S.-F. Chang, "Semi-Fragile Watermarking for Authenticating JPEG Visual Content", *SPIE Security and Watermarking of Multimedia Contents II EI '00*, Jan. 2000.

[7] Q.-B. Sun, Q. Tian and S.-F. Chang, "A robust and secure media signature scheme for JPEG images", *MMSP02*, Dec. USA.

[8] W. Zhu, K. H. Yang and M. J. Beachen, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs Technical Journal*, pp. 21-29, July-Sept., 1998.

[9] Q.-B. Sun, T.-T. Ng and Q. Tian, "Compressed-domain watermarking surviving CIF-to-QCIF conversion", *submitted to PCM2003*.

[10] W.-N. Lie, G.-S. Lin and T.-C.,Wang, "Digital watermarking for object-based compressed video", *ISCAS2001*, pp. 49 –52, 2001.