A Secure and Robust Authentication Scheme for Video Transcoding

Qibin Sun, Dajun He, and Qi Tian, *Senior Member, IEEE*

***Abstract*—In this paper, we describe a configurable content-based MPEG video authentication scheme, which is robust to typical video transcoding approaches, namely frame resizing, frame dropping and requantization. By exploiting the synergy between cryptographic signature, forward error correction (FEC) and digital watermarking, the generated content-based message authentication code (MAC or keyed crypto hash) is embedded back into the video to reduce the transmission cost. The proposed scheme is secure against malicious attacks such as video frame insertion and alteration. System robustness and security are balanced in a configurable way (i.e., more robust the system is, less secure the system will be). Compressed-domain process makes the scheme computationally efficient. Furthermore, the proposed scheme is compliant with state-of-the-art public key infrastructure. Experimental results demonstrate the validity of the proposed scheme.**

***Index Terms*—Digital signature, forward error correction (FEC), message authentication code (MAC), scalable video authentication, watermarking.**

I. Introduction

IN VIDEO surveillance or other legitimacy related applications, authentication of the transmitted video is usually required. For example, in the recent Bali bombing trial against Ba'asyir, who is the main suspect for planning the bombing, Indonesian prosecutors used testimony delivered by videophone from alleged militants jailed in Singapore.[1] In such cases, the whole video network must guarantee that the true origin of the transmitted video is Singapore and nothing in the video is altered during the transmission between Singapore and Indonesia in order to convince the judges. Such requirements are actually well aligned with the definition of the term "authentication" in cryptography which means both protecting the video integrity and preventing repudiation from the video sender [1].

Crypto signature techniques (e.g., public key based digital signature) [1] are a natural solution for addressing such authentication problems, assuming that no distortion is introduced during the video transmission. Given a video with arbitrary size, applying crypto hashing on the video to obtain its message authentication code (MAC) which is usually hundreds bits in length (e.g., 128 bits with MD5 algorithm and 160 bits with SHA-1 algorithm [1]). Signing on the MAC to generate the crypto signature of the video by using the sender's private key, and sending the video together with the signature to the recipient. At the receiver site, the authenticity of the video is verified through the following steps: Applying the same hash function, as used at the sending site, to obtain a MAC A. Decrypting the received signature by using the sender's public key to obtain another MAC B. Comparing A and B bit by bit: the received video will be deemed unauthentic if any discrepancies, even one bit difference, occur.

However, in real applications of video streaming over the networks, the video to be sent is often required to be transcoded in order to adapt to various channel capacities (e.g., network bandwidth) as well as terminal capacities (e.g., computing and display power) [2]. Throughout this paper, we essentially regard transcoding as the process of converting a compressed bitstream into lower rates without modifying its original structure [3]–[5]. Such transcoding poses new challenges on authentication due to, (1) the distortions introduced during video transcoding and (2) flexibilities of various video transcoding methods. It therefore demands a practical video authentication solution that differentiates malicious attacks from acceptable manipulations in video transcoding. The objective of this paper is to study this new problem and propose a secure and robust (i.e., semifragile) authentication system for adaptive video transmission.

In this paper, we present a content-based video authentication system, which is robust to frame resizing, frame dropping, requantization and their combinations. The scheme achieves an end-to-end authentication that is independent of specific transcoding design and balances the system performance in a configurable way. Furthermore, compressed-domain operation is adopted to reduce the computation cost of video signing and verification. Digital watermarking is employed to reduce the transmission cost of the signed video. The proposed solution is compliant with public key infrastructure (PKI) except that the video to be transmitted is watermarked.

The rest of this paper is organized as follows. Section II introduces the related techniques on cryptographic streaming authentication, video streaming and transcoding, and semifragile video integrity protection. Section III describes the proposed solutions that are robust to requantization, frame dropping. In Section IV, we continue discussing our authentication and watermarking solution robust to CIF-to-QCIF conversion. Experimental results are given in Section V. Conclusions and future work are given in Section VI.

Manuscript received November 25, 2004; revised December 2, 2005. This paper was recommended by Associate Editor C. W. Chen.

Q. Sun and Q. Tan are with the Institute for Infocomm Research, Singapore 119613 (e-mail: qibin@i2r.a-star.edu.sg; tian@i2r.a-star.edu.sg).

D. J. He was with the Institute for Infocomm Research, Singapore 119613. He is now with Shanghai Zhangjiang (Group) Company Ltd., Shanghai 201203, China.

Color versions of Figs. 9–12 and 14–16 are available at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2006.882540

[1]CNN News. [Online]. Available: http://www.cnn.com/2003/WORLD/asiapcf/southeast/08/21/trial.cleric.ap/

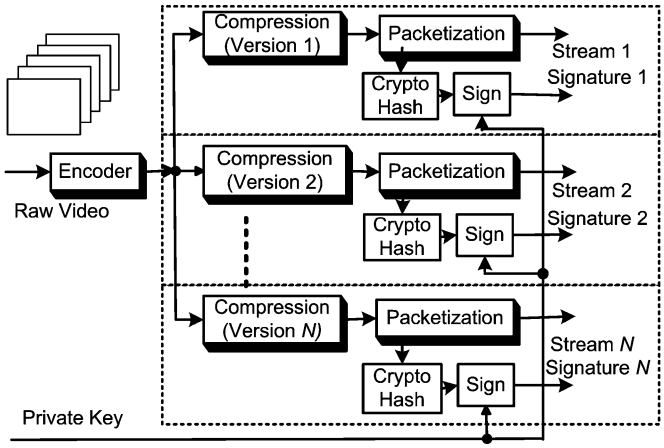

Fig. 1. Simple solution for signing precoded video streams.

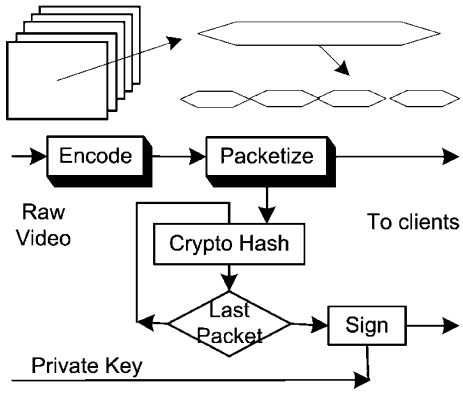

Fig. 2. Practical solution for stream signing.

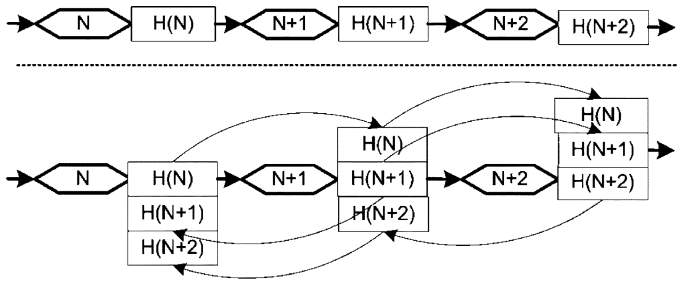

Fig. 3. Stream authentication resilient to packet loss.

II. Related Prior Techniques

A. Video Stream Authentication Based on Cryptographic Techniques

Assuming video streaming does not need to dynamically adapt to channel and terminal conditions, then we can preencode the video into several versions with different compression ratios. Different versions of the video could be sent out based on the requests from different recipients. For example, some recipients may want the compressed video with 1 Mb/s, while others may only want the video with 64 kb/s. In this scenario, video authentication also becomes simple: We directly employ a typical crypto signature scheme such as RSA or DSA [1], as illustrated in Fig. 1. The sender uses its private key to sign on the crypto hashes (i.e., MACs) of different versions of the compressed video and generates their corresponding signatures. The signature, together with the video, is sent to the recipients in order to prove the authenticity of the video at the receiver site, by using the sender's public key. Considering that the video may need to be verified part by part, a group of signatures could be obtained by partitioning the video into various parts before signing on them.

Signature generation is usually more time-consuming than its verification. In order to reduce the system computation, a typical video authentication system only signs on the last group of video packets instead of signing on packets group by group, as illustrated in Fig. 2 [6]. The MAC of every group is hashed with the MACs from its previous groups. The sender's private key signs on the MAC of the last group to form the signature of this video. At the receiver site, the recipient repeats the same operation as that at the sender site. The authenticity of the whole video can then be verified after the recipient receives the signature and the last group of video packets, by using the sender's public key. Yu [7] has successfully applied this idea to authenticate scalable media.

When the video is streamed over unreliable channels or protocols such as wireless or user datagram protocol (UDP), some packets may be lost during streaming. To overcome this problem, various approaches based on the concept of FEC [8] are proposed (Refer to Fig. 3). The basic idea is to append several MACs (i.e., H) from other packets to the current transmitted packet: If the current packet (e.g., N) is lost, its MAC can be correctly obtained from other packets (e.g., $N+m$) so that signature verification on the whole video can still be carried out. Obviously, these solutions will result in an extra transmission cost which depends on the rate of packet loss. Report has shown that [8], given a packet size of 128 bytes, packet loss rate of 0.2, verification rate of 100%, and hash size of 16 bytes, the transmission overhead will be around 100 bytes per packet, which almost doubles the original transmission cost. Such a high overhead is not acceptable for most video streaming applications. Another related problem is that, in compressed video bitstream, different packets may not be of the same importance. For instance, packets containing dc components are more important than those containing only ac components. The unequal importance of packets will also make the system design [e.g., the forward error correction (FEC) scheme] much more complicated. The last, yet the most critical problem might be the unstable data caused by video transcoding, because it could make the transcoded bitstream totally different from its original in terms of the data (binary) representation. Since crypto based authentication algorithms act on specific representation of data, the transcoded video with data represented in different ways could result in a failure of these authentication techniques.

B. Brief Introduction to Video Streaming and Transcoding

With the move towards convergence of wireless, Internet, and multimedia, the scalability of video coding and streaming becomes increasingly important for rich media access from anywhere, by anyone, at any time, with any device, and in any

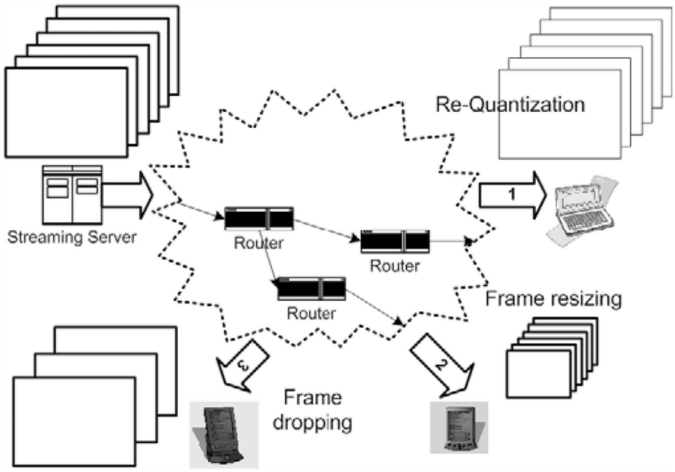

Fig. 4. Typical transcoding methods for scalable video streaming.

form [2]. Such scalability is typically accomplished by providing multiple versions of a video, in terms of signal-to-noise (SNR) scalability, spatial scalability, temporal scalability, or combinations of these options. It can be done either during video encoding (such techniques are called scalable video coding [9]) or during video streaming (such techniques are called transcoding [2]–[5]).

In case of authenticating a scalable compressed video such as MPEG4-FGS, we could employ the solution based on our previous work on JPEG2000 authentication [10]. In this paper we only focus on authenticating transcoded video, as illustrated in Fig. 4. A video is compressed and stored in the streaming server at bitrate a. Assuming that a terminal can only consume the video at bitrate b $(b < a)$, a transcoder is therefore required to convert the video from bitrate a to bitrate b. Three common video transcoding approaches are usually used: frame resizing, frame dropping and requantization [2]–[5]. Typically, they are performed in compressed domain to reduce the computation cost [11], [12].

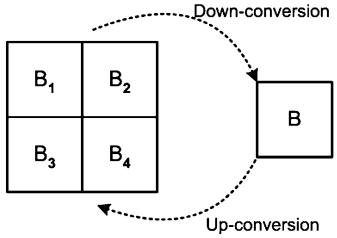

Fig. 5. Relationship between 4 blocks in CIF and one block in QCIF.

Frame Resizing: The first way to transcode video is spatial resolution down-conversion. One example is to convert the video from CIF to QCIF which resizes a video from the frame size of 352×288 to the frame size of 176×144 [11]. The core part of such transcoding is the down-conversion operation, which is performed normally in DCT domain. As shown in Fig. 5, given four 8×8 adjacent DCT blocks $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$, one new 8×8 DCT block $\mathbf{B}$ is generated by

$$\mathbf{B} = \text{down}\left(\begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{pmatrix}\right) = \mathbf{M}\begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{B}_3 & \mathbf{B}_4 \end{pmatrix}\mathbf{M}^T \quad (1)$$

where $\mathbf{M} = (\mathbf{M}_1\ \mathbf{M}_2)$ and M_1 and M_2 are given as (2) and (3), shown at the bottom of the page [11].

After frame down-scaling, the motion vectors (MVs) also need to be scaled. We skip its technical description here, interested readers please refer to [11], [12] for more details.

Frame Dropping: The second way to transcode video is the temporal resolution reduction or frame rate reduction. The direct way is by frame dropping. For example, in a compressed MPEG1/2 video, the video with frame rate a could be transcoded into a new video with frame rate $b(b < a)$ by directly dropping the B- or P-frames, after partially decoding the compressed video bitstream [5].

Requantization: The third way to transcode video is quality reduction or SNR reduction. Given a compressed video without

$$\mathbf{M}_1 = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.453 & 0.208 & -0.037 & 0.011 & 0 & -0.011 & 0.037 & -0.208 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & -0.5 \\ -0.159 & 0.396 & 0.257 & -0.049 & 0 & 0.049 & -0.257 & -0.396 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & -0.5 & 0 \\ 0.106 & -0.176 & 0.384 & 0.245 & 0 & -0.245 & -0.384 & 0.176 \\ 0 & 0 & 0 & 0.5 & 0 & -0.5 & 0 & 0 \\ -0.09 & 0.139 & -0.188 & 0.433 & 0 & -0.433 & 0.188 & -0.139 \end{bmatrix} \quad (2)$$

$$\mathbf{M}_2 = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.453 & 0.208 & -0.037 & 0.011 & 0 & -0.011 & -0.037 & -0.208 \\ 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0.159 & 0.396 & -0.257 & -0.049 & 0 & 0.049 & 0.257 & -0.396 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & -0.5 & 0 \\ -0.106 & -0.176 & -0.384 & 0.245 & 0 & -0.245 & 0.384 & 0.176 \\ 0 & 0 & 0 & -0.5 & 0 & 0.5 & 0 & 0 \\ -0.09 & 0.139 & 0.188 & 0.433 & 0 & -0.433 & -0.188 & -0.139 \end{bmatrix} \quad (3)$$

coded SNR scalability (e.g., MPEG1/2), its quality reduction could be done by partially decoding the compressed video stream to discrete cosine transform (DCT) domain, requantizing the DCT coefficients with a larger quantization step size and finally reencoding them by an entropy coder. In this paper, we consider dropping high frequency DCT coefficients [3] as a special case of requantization, by setting part of the elements in the quantization matrix to ∞.

In real applications, usually the video is transcoded by a combination of the above mentioned three approaches. From the viewpoint of authentication, we can see that video transcoding is actually a process of introducing incidental distortions (i.e., video quality degradation). These distortions usually result in a failure of directly applying crypto based authentication techniques to video transcoding applications [6]–[8].

C. Semifragile Video Integrity Protection

Digital watermarking is known as a good solution for content integrity protection by transparently and robustly embedding the secret data into the content [13]–[15]. But without introducing other sophisticated mechanism, watermarking itself is unable to prove who actually embeds the watermark (i.e., the source identification) because, usually the same key is used for both watermark embedding and watermark extraction. Stemmed from crypto signature techniques, the content-based signature schemes have been proposed for robust image and video authentication [16], [17]. Their solutions take the advantages of the invariant features extracted from the content and generate the content-based robust signature. Though signature based robust authentication schemes are able to protect both the content integrity and identify the content sender, the extra payload of the generated signature is still a problem because the size of the generated signature is usually proportional to the size of the original content.

In next sections, we shall study the authentication issues of video transcoding [18] and propose our solutions after reviewing the principles of invariant feature extraction and watermarking proposed in [15]–[17].

III. System Overview and the Proposed Schemes to ReQuantization and Frame-Dropping

A. General Requirements and System Overview

A typical application scenario for video streaming and transcoding is illustrated in Fig. 4. Considering the variations in the transmission channels and the terminals, we would argue that a robust and secure authentication scheme for video transcoding should satisfy the following prerequisites.

- **Robustness:** The authentication scheme must be robust to the video transcoding approaches, namely video requantization, frame resizing, frame dropping, and their combinations.
- **Security:** The authentication scheme must be secure enough to prevent the possible malicious attacks such as frame insertion/removal/replacement, or some in-frame modifications (e.g., content copy-paste) which intend to change the meaning of the video content.

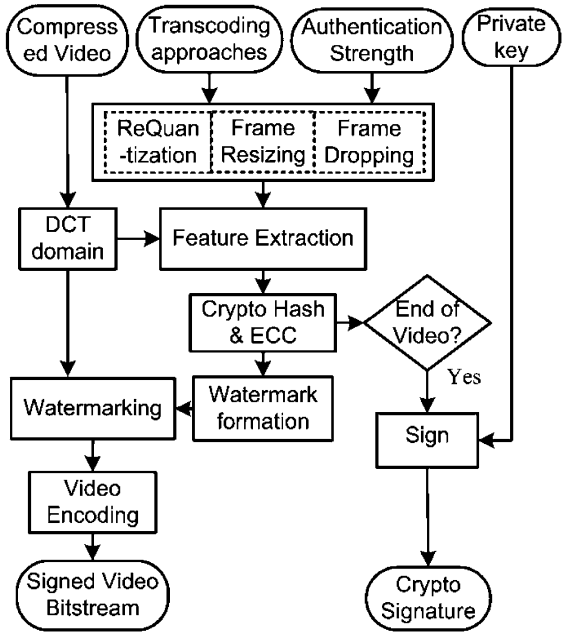

Fig. 6. System diagram of the proposed authentication solution robust to video transcoding.

- **Efficiency:** The authentication scheme should be very efficient. This is especially important for video applications because of its computation complexity and transmission cost.
- **Independency:** The authentication scheme should be independent of the specific network infrastructure and protocol used for video streaming. For example, considering that the coded video could be transcoded either at the server site before streaming or at some intermediate network nodes (e.g., routers) during streaming, this would make end-to-end authentication a requirement for the scheme to achieve.

A system diagram of our proposed system (signing part) is shown in Fig. 6. The signing operations are performed in the DCT domain to reduce system computation complexity. With reference to Fig. 6, three inputs for video signing are: the video sender's private key, the authentication strength, and possible transcoding approaches such as frame dropping, resizing, requantization or a combination of them. Here the authentication strength means protecting the video content to a certain degree (i.e., the video will not be deemed as authentic if it is transcoded beyond this degree). In this paper, we mainly use the quantization step size to control the authentication strength [15]–[17]. Based on the given transcoding approaches and the authentication strength, we extract the invariant features from DCT coefficients. Such frame-based features are cryptographically hashed, concatenated and then coded by a FEC scheme and embedded back into the DCT domain as a watermark. Note that the selected watermarking scheme is also required to be robust to the predefined transcoding approaches as well as the authentication strength. The watermarked video content is entropy coded again to form the signed video bitstream. In addition, the crypto hashing is recursively operated frame by frame, till the end of the video. The video sender's private key is used to sign on the

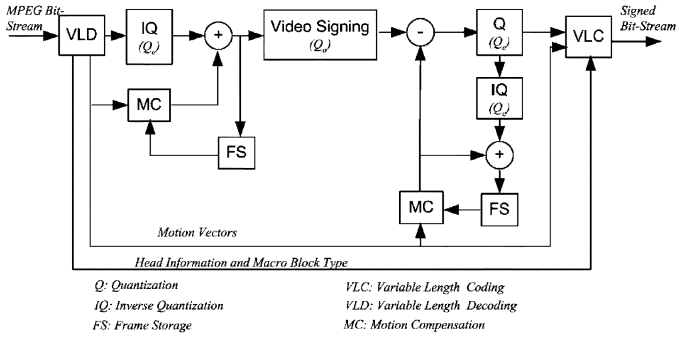

Fig. 7. Illustration of the video signing process.

final hash value to generate the crypto signature of this video. The signature, together with the watermarked video, is sent to the recipient to prove the authenticity of the video. At the receiver site, the verification of video authenticity is actually an inverse process of video signing by using the video sender's public key.

The video signing and verification are both performed in the compressed domain to reduce system computation, as shown in Fig. 7. The input MPEG bitstream is partially decoded to obtain DCT coefficients. The video signing operation (i.e., feature extraction, watermarking and crypto hashing etc.) is then performed on the DCT coefficients, frame by frame. Finally, the signed video is reencoded using the MVs from the original MPEG bit stream. Note that during the whole process we keep the MVs unchanged to avoid motion reestimation, although such a process may slightly degrade the quality of the newly coded video, because the watermarked DCT coefficients are different from the original DCT coefficients used for estimating the MVs. However, motion estimation is very time-consuming. Hence, skipping it (Fig. 7) will greatly reduce system computation. As pointed out in [13] that, watermarking may cause a drift effect in the coded video that is visually annoying. The scheme shown in Fig. 7 also has the function on drift compensation. It means that the distortion caused by watermarking will be "compensated" by reencoding the B- and P-frames.

Considering the fact that different transcoding approaches affect the robustness of the extracted feature and watermarking in different ways, we shall discuss them separately. Note that, if the video transcoding is a combination of the above-mentioned three approaches, the selected features should be an intersection of each extracted feature set. In order to describe our proposed authentication system (Figs. 6 and 7) in a clear way, we shall introduce our countermeasures to address the robustness to these three transcoding approaches one by one, starting from the solution to quantization-based transcoding, followed by frame-dropping based transcoding and finally frame-resizing based transcoding.

B. Authentication Robust to Quantization-Based Transcoding

The first transcoding approach is to requantize the DCT coefficients with a larger quantization step size as increasing quantization step size will result in a bit rate reduction [3].

Making the extracted feature and watermarks robust to requantization in transform domain (e.g., DCT) has been thoroughly studied in [15], [16] where the authors explore two invariant properties of quantization based lossy compression (e.g., JPEG) for image authentication. The first property is the invariant relationship between two coefficients in a block pair before and after JPEG compression and is used for extracting the features. The second one shows that if a DCT coefficient is modified to an integral multiple of a quantization step size Q_a, which is larger than the steps size Q_c (i.e., $Q_a > Q_c$) used in later JPEG compression, then this coefficient can be exactly reconstructed after later JPEG compression. This invariant property is unambiguous and used for watermark embedding. For instance, if the watermark bit is "zero," then the coefficient quantized by Q_a should be even or modified to be even by adding or subtracting a "one." If the watermark bit is "one," then the coefficient quantized by Q_a should be odd or modified to be odd by adding or subtracting a "one." This watermarked coefficient is de-quantized by Q_a and then quantized by Q_c again in subsequent compression.

In our proposed solution for video authentication, we only employ the second property for both feature extraction and watermarking[2] because its exact reconstruction on the quantized value makes the crypto hashing workable in our specific application where the distortions could be introduced during transcoding. For MPEG video, such invariance is naturally maintained for I-frame because compressing I-frame is the same as compressing JPEG image. The interesting thing is, we also found that this invariant property is also kept for P- and B-frames in MPEG bitstream under the assumption that the same predictive coefficient $\tilde{C}_i$ could be acquired during video coding and decoding. Actually, such assumption is valid for the scheme shown in Fig. 7 where drift compensation has been implemented. We give the proof[3] below.

Proof: Let the original DCT coefficient be C_p and the decoded DCT coefficient be C'_p. As we have assumed that the same predictive coefficients are used for video coding and decoding and C_p is a multiple of Q_a, so C'_p can be expressed as

$$\begin{aligned} C'_p &= \text{round}\left(\left(C_p - \tilde{C}_i\right)/Q_c\right) * Q_c + \tilde{C}_i \\ &= C_p - \tilde{C}_i + \Delta * Q_c + \tilde{C}_i \\ &= C_p + \Delta * Q_c \end{aligned} \tag{4}$$

where $|\Delta| \leq 0.5$. Since $C_p = n * Q_a$, we have

$$\begin{aligned} \text{round}\left(C'_p/Q_a\right) &= \text{round}((C_p + \Delta * Q_c)/Q_a) \\ &= n + \text{round}((\Delta * Q_c)/Q_a) \\ &= n \\ &= \text{round}(C_p/Q_a). \end{aligned} \tag{5}$$

The above proof shows that the second invariant property in [15] and [16] can also be extended from JPEG image or MPEG I-frame to MPEG B- or P-frames.

[2]Note that we do not employ watermarking in this subsection (the solution robust to quantization-based transcoding), but we shall use it in the solutions robust to frame-dropping and frame-resizing transcoding.

[3]For simplicity, here we do not consider the case of "dead-zone" in MPEG coding.

Video signing algorithm robust to quantization-based transcoding is given below (the video is assumed to be coded in variable bit rate (VBR) mode where the quantizer step size is kept constant across frames).

Algorithm 1.a. Video signing (robust to quantization-based transcoding)

Input

Video sender's private key *Pri*.

Original video V_o (V frames) to be protected, it may undergo the quantization-based transcoding.

Authentication quantization step size Q_a.

MPEG compression quantization step size $Q_c (Q_c < Q_a)$.

Begin

For $k = 0 : V - 1$ **Do** // *Loop on video frames*

Decode the video bitstream to a number (N) of 8×8 DCT blocks, frame by frame.

Label all DCT coefficients in zig-zag order, denoted as: $\{D_{ij}^O : 0 \le i < 64; \;\; 0 \le j < N\}$.

Select dc coefficients from all blocks to form feature set $F_k : F_k = \{D_{0j}^O; \quad 0 \le j < N\}_k$.

Quantize F_k by Q_a, obtain $\bar{F}_k$.

Crypto hash $\bar{F}_k$, obtain

$$H_k = \begin{cases} h(\bar{F}_k), & k = 0 \\ h(\bar{F}_k, H_{k-1}), & k > 0. \end{cases}$$

Dequantize $\bar{F}_k$ by Q_a, obtain F_k'.

Quantize F_k' by Q_c, entropy coding to generate MPEG bitstream while keeping MV unchanged.

End

Sign on H_k by *Pri* and obtain the signature G.

End

Output

Recompressed video V_w.

Content based signature G.

Algorithm 1.b. Video verifying (robust to quantization-based transcoding)

Input

Video sender's public key *Pub*.

The video V_w (V frames) to be authenticated, it may undergo the quantization-based transcoding.

Authentication quantization step size Q_a.

MPEG compression quantization step size $Q_c (Q_c < Q_a)$.

Content based signature G.

Begin

For $k = 0 : V - 1$ **Do** // *Loop on video frames*

Decode the video bitstream to a number (N) of 8×8 DCT blocks, frame by frame.

Label all DCT coefficients in zig-zag order, denoted as: $\{D_{ij}^O : 0 \le i < 64; \quad 0 \le j < N\}$.

Select dc coefficients from all blocks to form feature set $F_k : F_k = \{D_{0j}^O; \quad 0 \le j < N\}_k$.

Quantize F_k by Q_a, obtain $\bar{F}_k$.

Crypto hash $\bar{F}_k$, obtain

$$H_k = \begin{cases} h(\bar{F}_k), & k = 0 \\ h(\bar{F}_k, H_{k-1}), & k > 0 \end{cases}$$

End

Decrypt the signature G by *Pub* to obtain H_k'.

$\mathrm{Auth} = H_k \oplus H_k'$

End

Output

If $\mathrm{Auth} = 0$, the video is authentic; **Else** the video is unauthentic

The above algorithm is very similar to the one shown in Fig. 2 except that hashing operation is applied to video frame level as opposed to transport packet level. In the case that only quantization based transcoding is employed, we can skip watermarking because no side information is required to be transmitted to the terminals for authentication purpose.

C. Authentication Robust to Frame Dropping Based Transcoding

The second acceptable transcoding for bit rate reduction is frame dropping. For instance, the original video sequence "Salesman" encoded at 64 kb/s with 30 frames per second (fps) can be transcoded to a new version of 32 kb/s by dropping to 10 fps [5]. In order to have low computation, low memory usage and high visual quality, the state-of-the-art frame-dropping based video transcoding is usually performed in compressed-domain and the frames are dropped in a flexible way. For instance, an original video with the frames like $\mathrm{I_1B_2B_3P_4B_5B_6P_7B_8B_9P_{10}B_{11}B_{12}I_{13}}$ could be transcoded to a new one whose frames are $\mathrm{I_1B_2P_4B_5P_7B_8P_{10}B_{11}I_{13}}$ (i.e., linear dropping) or $\mathrm{I_1B_2B_3P_4B_5B_6P_7B_8P_{10}I_{13}}$ (i.e., nonlinear dropping). Therefore, the proposed robust video authentication solution should meet these transcoding requirements. It means, if the frames of a video are received incomplete only because of transcoding, we would still like to be able to authenticate

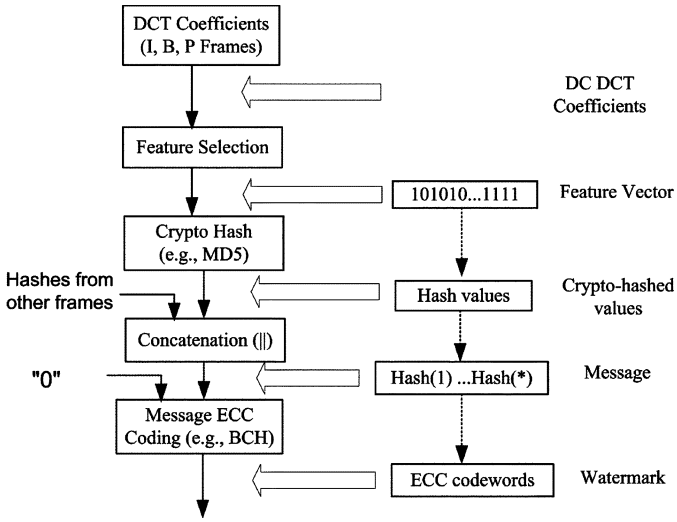

Fig. 8. Authentication robust to frame-dropping.

this video as if it is the same as the original one whose frames were not lost. This defines resistance to frame loss in a strong sense: a video frame is either dropped or authenticable.

We solved this problem based on the concept of FEC, which is similar to what is illustrated in Fig. 3, where the idea is to pay extra transmission cost (i.e., multiple times of transmitted crypto hashes) for authentication robustness. We further resolve this extra payload problem by embedding those hash values generated from other frames into current frame using watermarking. If some frames are dropped during transcoding, we can obtain their corresponding crypto hashes from other frames and the whole authentication process for other frames can still be continuously executed.

The process flow is illustrated in Fig. 8, which details Fig. 6 specifically for authentication robust to frame dropping. In (4) and (5), we have proven that similar to I-frames, there is also an invariant property in B- and P-frames. Therefore, for all video frames, we adopt the same feature extraction mechanism as the one for quantization based transcoding. For the purpose of system simplicity, within one group of pictures (GOP), we may not need to directly extract the feature from other frames (i.e., B-or P-). Instead, we could take its I-frame feature plus the frame number within this GOP as the input to generate its corresponding crypto hash value. Note that such modification may affect system security especially when there is not too much coherence of the frames within one GOP. The hash is concatenated with the hashes from other frames. The combined hash value is FEC coded and embedded back to the current frame as watermark to achieve the authentication robustness to frame-dropping based transcoding. Note that due to the watermarking capacity, we may only embed part of the crypto hash of each frame. More sophisticated FEC schemes [8] could be employed to further improve not only system authentication robustness under the same watermarking payload, but also the flexibility of the way in which frame-dropping is applied (e.g., nonlinear).[4]

[4]We skipped the details of technical implementation in this paper due to the limits of paper length.

IV. Proposed Solution to CIF-to-QCIF Conversion

A. Authentication Robust to Frame-Resizing Based Transcoding

The third acceptable transcoding we consider is frame-resizing,[5] i.e., change the frame size down to a smaller one. For instance, the conversion of the video from CIF to QCIF corresponds to a change in frame size from 352×288 down to 176×144. To make the authentication robust to frame-resizing, it naturally requires the feature extraction and watermarking to survive such transcoding, which is preferably implemented in the DCT domain for the purpose of computational efficiency [11], [12].

To meet this robustness requirement, we could perform the feature extraction operation and watermarking in the QCIF domain instead of the CIF domain. However, because the watermarked video before streaming should still be in its original frame size (e.g., CIF) while the received video could be either in CIF or QCIF, we have to generate the watermarked CIF video at the server site. Although the watermarking schemes robust to geometrical distortions (e.g., scaling, rotation) have been proposed [19], we here propose a compressed-domain watermarking solution surviving CIF-to-QCIF conversion [20] based on the following considerations. Firstly, the desired scheme must be computationally efficient especially at the video verification (i.e., watermark extraction) site. Therefore, a fast, simple and compressed-domain watermarking scheme is desired. Similar to the digital signature scheme whose signature generation is usually much more time-consuming than signature verification [1], we could also tolerate the processing time of the video signing in our proposed scheme (mainly feature extraction, watermarking and signature signing) to be longer than that of the video verification. Secondly, the selected scheme should be well suited for our proposed video authentication scheme robust not only to frame resizing but also to frame dropping and requantization which are all performed in compressed domain.

Although both the original video and the signed video are in CIF, the actual watermark embedding and extraction are performed in QCIF regardless of the format of the video to be verified. We then map the difference between the "original" QCIF video frame and the watermarked QCIF video frame back to the CIF video frame to generate the watermarked CIF video (Note that we own both CIF video frame and QCIF video frame at the server/signing site). In the next section, we shall describe the proposed watermarking solution surviving CIF-to-QCIF conversion.

B. Basic Idea of the Proposed Watermarking Scheme

Let's go back to Fig. 5 and (1) again. Given four 8×8 adjacent DCT blocks $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$, one new 8×8 DCT block $\mathbf{B}$ can be obtained. Similarly, given DCT block $\mathbf{B}$, its corresponding four blocks $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$ can

[5]For the reason of simplicity, we only consider the case of CIF-to-QCIF in this paper. It can be extended to other frame-resizing approaches as long as the frame size is half scaled.

be approximately obtained by its up-conversion, which is an inverse process of down-conversion

$$\begin{pmatrix} \bar{\mathbf{B}}_1 & \bar{\mathbf{B}}_2 \\ \bar{\mathbf{B}}_3 & \bar{\mathbf{B}}_4 \end{pmatrix} = \text{up}(\mathbf{B}) = \mathbf{M}^{-1}\mathbf{B}(\mathbf{M}^T)^{-1} \tag{6}$$

where $(*)^{-1}$ denotes pseudo inverse of matrix (*).

Note that an up-conversion on $\mathbf{B}$ would only produce an approximated version of $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$, in a least-square sense,[6] in the DCT domain. This is because actually the down-conversion is a many-to-one mapping while the up-conversion is a one-to-many mapping. However, an up-conversion followed by a down-conversion leaves a matrix unchanged (i.e., $\mathbf{W}' = \text{down}(\text{up}(\mathbf{W})) \approx \mathbf{W}$). Such observation has made accurate watermark extraction possible for our watermarking scheme because we embed and extract the watermark in QCIF video instead of CIF video. We illustrate this observation as follows.

Assume that there is a small and independent perturbation $\mathbf{W}$ (watermarking) onto $\mathbf{B}$, we have

$$\tilde{\mathbf{B}} = \mathbf{B} + \mathbf{W}. \tag{7}$$

Then, according to (6), its up-conversion is given by

$$\begin{aligned} \text{up}(\tilde{\mathbf{B}}) &= (\mathbf{M})^{-1}(\mathbf{B}+\mathbf{W})(\mathbf{M}^T)^{-1} \\ &= (\mathbf{M})^{-1}\mathbf{B}(\mathbf{M}^T)^{-1} + (\mathbf{M})^{-1}\mathbf{W}(\mathbf{M}^T)^{-1} \\ &= \begin{pmatrix} \bar{\mathbf{B}}_1 & \bar{\mathbf{B}}_2 \\ \bar{\mathbf{B}}_3 & \bar{\mathbf{B}}_4 \end{pmatrix} + \begin{pmatrix} \bar{\mathbf{W}}_1 & \bar{\mathbf{W}}_2 \\ \bar{\mathbf{W}}_3 & \bar{\mathbf{W}}_4 \end{pmatrix} \\ &= \begin{pmatrix} \bar{\mathbf{B}}_1 + \bar{\mathbf{W}}_1 & \bar{\mathbf{B}}_2 + \bar{\mathbf{W}}_2 \\ \bar{\mathbf{B}}_3 + \bar{\mathbf{W}}_3 & \bar{\mathbf{B}}_4 + \bar{\mathbf{W}}_4 \end{pmatrix}. \end{aligned} \tag{8}$$

Similarly, assuming that a small and independent perturbation is added onto four blocks $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and $\mathbf{B}_4$, their down-conversion would be as follows:

$$\begin{aligned} &\text{down}\left(\begin{pmatrix} \bar{\mathbf{B}}_1 + \bar{\mathbf{W}}_1 & \bar{\mathbf{B}}_2 + \bar{\mathbf{W}}_2 \\ \bar{\mathbf{B}}_3 + \bar{\mathbf{W}}_3 & \bar{\mathbf{B}}_4 + \bar{\mathbf{W}}_4 \end{pmatrix}\right) \\ &= \mathbf{M}\begin{pmatrix} \bar{\mathbf{B}}_1 + \bar{\mathbf{W}}_1 & \bar{\mathbf{B}}_2 + \bar{\mathbf{W}}_2 \\ \bar{\mathbf{B}}_3 + \bar{\mathbf{W}}_3 & \bar{\mathbf{B}}_4 + \bar{\mathbf{W}}_4 \end{pmatrix}\mathbf{M}^T \\ &= \mathbf{M}\begin{pmatrix} \bar{\mathbf{B}}_1 & \bar{\mathbf{B}}_2 \\ \bar{\mathbf{B}}_3 & \bar{\mathbf{B}}_4 \end{pmatrix}\mathbf{M}^T + \mathbf{M}\begin{pmatrix} \bar{\mathbf{W}}_1 & \bar{\mathbf{W}}_2 \\ \bar{\mathbf{W}}_3 & \bar{\mathbf{W}}_4 \end{pmatrix}\mathbf{M}^T \\ &= \mathbf{M}(\mathbf{M})^{-1}\mathbf{B}(\mathbf{M}^T)^{-1}\mathbf{M}^T + \mathbf{M}(\mathbf{M})^{-1}\mathbf{W}(\mathbf{M}^T)^{-1}\mathbf{M}^T \\ &= \mathbf{B} + \mathbf{W}. \end{aligned} \tag{9}$$

From (8) and (9), we can find such a fact: if a watermark is embedded in the QCIF, this watermark can still be extracted after the watermarked video is processed by up-conversion followed by down conversion. Equations (8) and (9) form the basis of our proposed algorithm for watermark embedding and extraction.

C. Watermark Embedding

As indicated in [15] and [16], the semifragile watermark should be embedded in the low or middle frequency DCT coefficients. Furthermore, to enhance the system robustness in our authentication system, it is better to separate the DCT coefficients for feature extraction from those for watermarking. If we select the dc coefficient as the feature, it means any modification (watermarking) on the DCT coefficients in QCIF video frame should not lead to a modification of the dc coefficient in the CIF video frame.

We could meet these watermarking criteria by carefully selecting the DCT coefficients in QCIF video frame for watermarking. Let's assume $\mathbf{M}^{-1} = [m_{ij}]_{16\times 8}$ and $(\mathbf{M}^T)^{-1} = [m'_{jk}]_{8\times 16}$. Then, the dc coefficient of block $\mathbf{B}_1$can be calculated according to

$$B_1(0,0) = \sum_{j=0}^{7}\sum_{k=0}^{7} m_{0,j}B(j,k)m'_{k,0} \tag{10}$$

Under the least square case, we have

$$\begin{cases} \mathbf{M}^{-1} = \mathbf{M}^T \\ (\mathbf{M}^T)^{-1} = \mathbf{M} \end{cases} \quad \text{and} \quad \begin{cases} m_{0,j} = 0, & \text{if } j = 2,4,6 \\ m'_{k,0} = 0, & \text{if } k = 2,4,6. \end{cases}$$

So, the dc coefficient in the CIF format will remain unchanged if we select the DCT coefficients, which are located in the second, fourth, and sixth rows (columns) in an 8×8 block of QCIF video frame, to embed watermark. As a result, we select the low-frequency DCT coefficients located in (0,2), (0,4), (2,0) and (4,0) to embed the watermark. Further, we only select those DCT coefficients with large magnitude for embedding, to improve the visual quality of the watermarked video frames. The watermark embedding in the QCIF can be described as following:

Algorithm 2. Watermark embedding into QCIF video frames

Begin

• The selected coefficients are quantized by the authentication strength Q_a.

• The quantized DCT coefficients are divided into two groups. Group 1 consists of all (0,2) and (2,0) coefficients in a video frame; and Group 2 consists of all (0,4) and (4,0) in a video frame. Every 3 randomly selected coefficients in Group 1 are grouped into one subgroup; and every 6 randomly selected coefficients in Group2 are grouped into one subgroup. Therefore, for a QCIF (176×144) video frame, we have $264 + 132 = 396$ subgroups.

• In every subgroup, the coefficient with the largest magnitude is selected to embed the watermark. If all coefficients are zero, the first coefficient in the subgroup is selected.

• Every selected coefficient is modified with the rule: If the watermark bit is "1," the coefficient is modified to be an even number; If the watermark bit is "0," the coefficient is modified to be an odd number.

• The watermarked DCT coefficients are inversely quantized by Q_a.

End

[6]The term least square describes a frequently used approach to solving overdetermined equations in an approximate sense. Instead of solving the equations exactly, we seek only to minimize the sum of the squares of the residuals.

Note that the format of the signed video is CIF, the watermarked QCIF video frame should be converted back to CIF before reencoding. Because the direct up-conversion from the QCIF frame to the CIF frame will seriously degrade visual quality, here we propose an alternative solution: the DCT coefficients in CIF video frame are modified according to the difference between the watermarked QCIF frame and the original QCIF frame. Let's take the DCT coefficient (0, 2) in QCIF as an example to study how this difference affects the coefficients in CIF. From (1), we have

$$\begin{aligned} B(2,0) &= 0.5(M(2,1)B_1(1,0) + M(2,1)B_2(1,0) \\ &\quad + M(2,7)B_1(7,0) + M(2,7)B_2(7,0) \\ &\quad + M(2,9)B_3(1,0) + M(2,9)B_4(1,0) \\ &\quad + M(2,15)B_3(7,0) + M(2,15)B_4(7,0)) \\ &= 0.25 * 0.9808(B_1(1,0) \\ &\quad + B_2(1,0) - B_3(1,0) - B_4(1,0)) \\ &\quad + 0.25 * 0.1951(-B_1(7,0) - B_2(7,0) \\ &\quad + B_3(7,0) + B_4(7,0)) \\ &= D_1 + D_2. \end{aligned} \tag{11}$$

If we set $B_1(7,0) = B_2(7,0) = B_3(7,0) = B_4(7,0) = 0$, then $D_2 = 0$. Equation (11) can be rewritten as (12). Note that all these 4 coefficients are high frequency coefficients. Hence, this setting (i.e., $D_2 = 0$) will not significantly degrade the video quality

$$\begin{aligned} B(2,0) &= D_1 = 0.9808 \\ &\quad * \frac{(B_1(1,0) + B_2(1,0) - B_3(1,0) - B_4(1,0))}{4}. \end{aligned} \tag{12}$$

Suppose $B^{'}(2,0)$ is the watermarked DCT coefficient in QCIF; and$B_1^{'}(1,0), B_2^{'}(1,0), B_3^{'}(1,0), B_4^{'}(1,0)$ are its corresponding DCT coefficients in CIF. Assume

$$B^{'}(2,0) = B(2,0) + W(2,0) \tag{13}$$

where $W(2,0)$ is the corresponding watermark value.

From (12), we can see that (13) will remain the same as long as (14) is satisfied, regardless of arbitrary modification on $B_1(1,0), B_2(1,0), B_3(1,0)$, and $B_4(1,0)$

$$\begin{aligned} &(B_1^{'}(1,0) + B_2^{'}(1,0) - B_3^{'}(1,0) - B_4^{'}(1,0)) - (B_1(1,0) \\ &\quad + B_2(1,0) - B_3(1,0) - B_4(1,0)) \\ &\quad = \frac{4 * W(2,0)}{0.9808} = 4 * \Delta^{'} \end{aligned} \tag{14}$$

where

$$\Delta^{'} = \frac{W(2,0)}{0.9808}. \tag{15}$$

Equation (14) tells us how to modify the DCT coefficients in CIF according to the difference between the original QCIF frame and the watermarked QCIF frame to achieve a better visual quality of the watermarked CIF frame. The relationships between the other selected coefficients in QCIF and their corresponding coefficients in CIF are listed as follows.

$$\begin{cases} B(0,2) = 0.9808 * \frac{(B_1(0,1) - B_2(0,1) + B_3(0,1) - B_4(0,1))}{4} \\ \Delta^{'} = \frac{W(0,2)}{0.9808} \end{cases} \tag{16}$$

$$\begin{cases} B(0,4) = 0.9239 * \frac{(B_1(0,2) + B_2(0,2) + B_3(0,2) + B_4(0,2))}{4} \\ \Delta^{'} = \frac{W(0,4)}{0.9239} \end{cases} \tag{17}$$

$$\begin{cases} B(4,0) = 0.9239 * \frac{(B_1(2,0) + B_2(2,0) + B_3(2,0) + B_4(2,0))}{4} \\ \Delta^{'} = \frac{W(4,0)}{0.9239} \end{cases}. \tag{18}$$

The process of mapping the difference from QCIF to CIF is described as follows.

Algorithm 3. Difference mapping from QCIF to CIF

Begin

• Set coefficients (0,6), (0,7), (6,0), and (7,0) to zero in the whole video frame to ensure that D_2 is equal to zero. Let F represent a modified video frame.

• Convert CIF video frame F to QCIF video frame QF.

• Embed the watermark in $QF$to get watermarked video frame QF_w.

• Calculate the difference between QF and QF_w : $QF_d = QF - QF_w$.

• Check the selected coefficients in QF_d. For any nonzero coefficients selected in QF_d, modify the magnitude of its corresponding coefficients in CIF according to the following criteria. Suppose b_1, b_2, b_3, and b_4 are the corresponding coefficients in B_1, B_2, B_3, and B_4, and $\Delta^{'}$ is the difference calculated according to (15)–(18). Modify the magnitude of corresponding coefficients $b_i$according to (19). The sign of $b_i$remains unchanged

$$|b_i^{'}| = \frac{|b_i|}{|b_1| + |b_2| + |b_3| + |b_4|} * 4 * \Delta^{'} \quad i = 1,2,3,4. \tag{19}$$

End

D. Watermark Extraction

Watermark extraction is performed in QCIF domain. (Note that if the video to be verified is CIF, CIF-to-QCIF conversion is needed.) Similar to the process of watermark embedding in QCIF, the selected coefficients are quantized by Q_a before we extract the watermark bit according to

$$\begin{cases} B(i,j) \text{ is even}, & \text{Watermark Bit} = 0 \\ B(i,j) \text{ is odd}, & \text{Watermark Bit} = 1. \end{cases} \tag{20}$$

Fig. 9. Test video sequences.

V. Experimental Results and Performance Analysis

A. Experimental Results

Five video sequences, Akiyo, Bike, Child, Coastguard and Salesman, (Fig. 9, from left to right), are used to evaluate the proposed system. All these five video sequences are encoded into MPEG1 bit stream by MPEG coding tool "TMPGEnc."[7] Constant quality (CQ), i.e., VBR mode, coding is chosen so that the quantizer step size keeps constant during the whole compression process. For example, the step size is 1 or 2 if the desired video coding quality is set to 70. We also assumed that quality 50, which corresponds to the quantizer step size of 6, is the lowest quality for our authentication scheme robust to multicycle compression. So the maximum of Q_a is set to 6 in our simulation. Therefore, Q_c must be equal or lower than Q_a. Note that in MPEG, both Q_c and the quantizer matrix are used to control the compressed video quality. Because we use the default qauntizer matrix specified by MPEG1 in our scheme, we shall only mention Q_c, which is used to control video quality. Note that in some practical systems, constant bit rate (CBR) is used for video coding where quantization step sizes vary across frames. In such case, we may only support the system robustness to the transcoding done by frame-dropping and CIF-QCIF conversion.

We firstly check the number of nonzero DCT coefficients in QCIF because if the number of nonzero coefficients is not sufficient, watermarking may cause serious quality degradation on video. Fig. 10 shows the ratio of the number of nonzero coefficients to the total number of 396 coefficients selected from 396 subgroups in one frame. The solid black line is the result of averaging the five videos along 300 frames. We can see that most of the selected coefficients are nonzero (The worst case is video "Akiyo" which still has about 84% nonzero coefficients). Therefore, our proposed watermarking scheme does not significantly degrade video quality while still keeping a high robustness (i.e., multicycle quantization and CIF-to-QCIF conversion) and high watermark capacity (i.e., 396 bits per frame).

Before designing the FEC coding scheme, the performance on watermark embedding and extraction should be studied. So a 396-bits watermark is embedded into every video frame first; then we extract the watermark from the watermarked video, which has been shown to pass multicycle compression or CIF-to-QCIF conversion. The performances are shown in Figs. 11 and 12, respectively. Fig. 11 is the result on the correctly extracted watermark under five rounds compression ($Q_a = 6$ and $Q_c <= 5$). We can see from most frames in all 5 videos that the percentages of correctly extracted watermark are above 96% which means only about 16 bits of errors occurred in all 396 embedded bits. There are only a few frames, whose percentage of correctly extracted watermark is lower than 96% but higher than 94% (24 bits of errors, video "Coastguard"). Fig. 12 is the result on the correctly extracted watermark under CIF-to-QCIF conversion ($Q_a = 3$ and $Q_c = 2$ (default)). The result is much better than Fig. 11: the lowest one is still above 97% (12 bits of errors). The cause of the errors could be as following. 1) Some approximation operations are employed during watermark embedding. For instance, setting D_2 to zero (11) and difference mapping from QCIF back to CIF (Algorithm 3). 2) Different implementations are used between video signature generation and video transcoding.

[7]Freeware MPEG Tool. [Online]. Available: http://www.tmpgenc.net/.

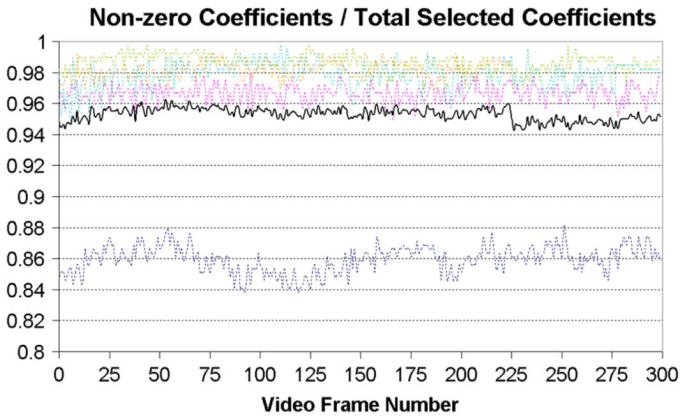

Fig. 10. Number of the selected DCT coefficients which are nonzero. Black solid line is the average result.

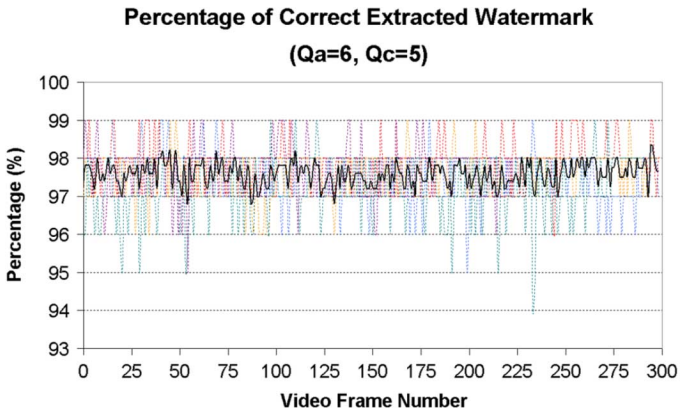

Fig. 11. Percentage of the correctly extracted watermark (Black solid line is the average result.). The watermark is 396 bits and the watermarked video ($Q_a = 6$) has undergone the multicycle quantization whose maximum step-size is set to 5.

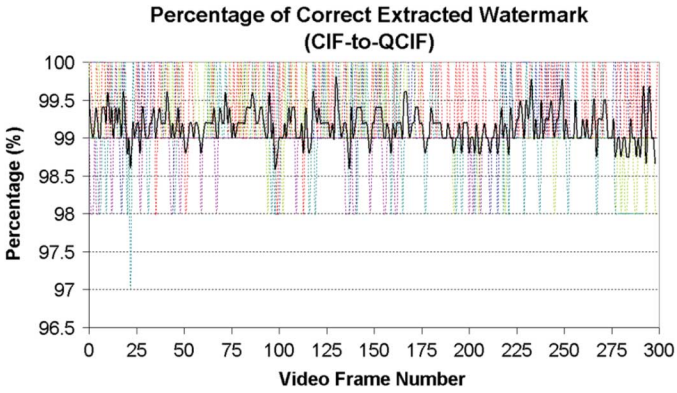

Fig. 12. Percentage of the correctly extracted watermark (Black solid line is the average result.). The watermark is 396 bits and the watermarked video ($Q_a = 3$ and $Q_c = 2$ (default)) has undergone the CIF-to-QCIF conversion.

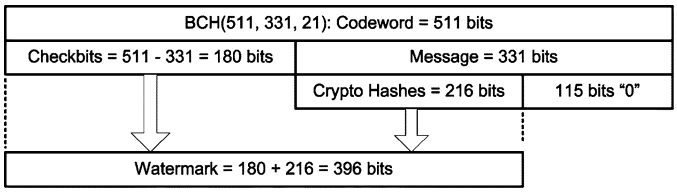

Fig. 13. Illustration on how to form the watermark bits.

Based on the above results on watermark bit-error rate (BER), we employed BCH (511, 331, 21), which is a binary error correction coding scheme named by its three inventors, Bose, Ray-Chaudhuri, and Hocquenghem, as the FEC coding scheme. It can correct up to 21 bits of errors among all 511 bits. However, the maximum watermarking capacity per video frame in our system design is 396 bits. Therefore, we have to modify the selected FEC scheme to make it fit into our application. Because we must embed 180 bits (i.e., $511 - 331 = 180$) parity check data, the room for embedding the crypto hash values is only $396 - 180 = 216$ bits. To be more precise, we padded 115 bits long message with "0" to create 331 (i.e., $216 + 115 = 331$) bits information for both FEC encoding and decoding, as shown in Fig. 13. The watermark therefore consists of 180 bits check data and 216 bits information bits.

The crypto hashing we adopted is MD5, which generates 128 bits of MAC given an arbitrary length of input message [1]. The 216 bits embedding capacity is enough for the authentication to be robust to requantization and CIF-to-QCIF. In the case of frame-dropping, we may have to embed only part of the crypto hash of each video frame because we also need to embed the crypto hashes from other video frames along with that of the current frame. For instance, if the authentication robustness is up to drop 5 frames (i.e., from 30 to 5 fps), it means we have to embed 6 frames' crypto hashes into the current video frame. Given the embeddable room of 216 bits per frame, we can only embed 36 bits of crypto hash of each frame. This is the maximum setting in our current prototype because short length of crypto hash may result in an easy break on the system.

The quality of the corresponding watermarked videos is shown in Fig. 14 in terms of Peak SNR (PSNR), measured under $Q_a = 6$ and $Q_c = 5$ (The performance of watermark detection is shown in Fig. 11). We can see the average PSNR

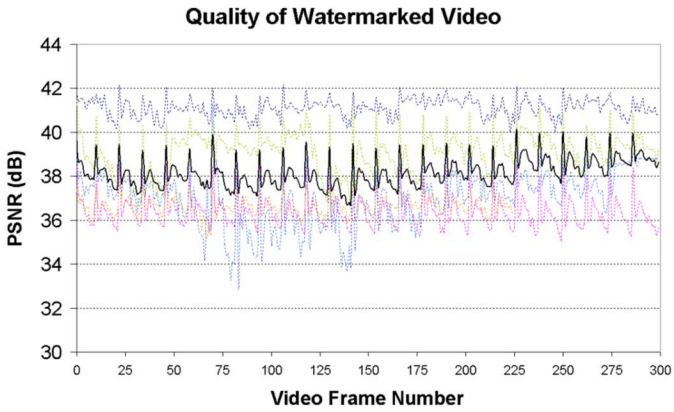

Fig. 14. Quality of six watermarked videos in terms of PSNR. Black solid line is the average result.

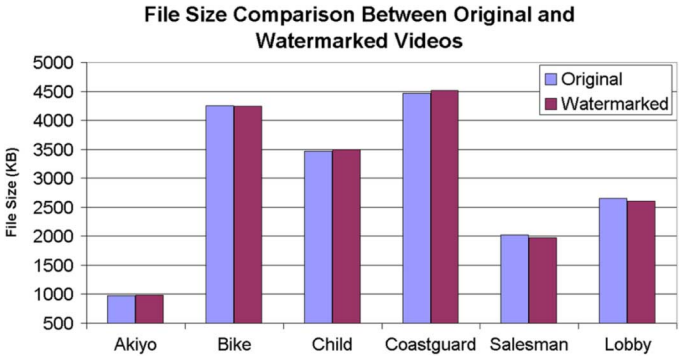

Fig. 15. Evaluation on file size change due to watermarking.

(black solid line) is above 37 dB (The worst case 33 dB was from "Coastguard"). Furthermore, the PSNR varies from frame to frame in each video. Usually the quality of watermarked I-frames is better than the B- or P-frames. This could be explained as following. 1). The distribution of DCT coefficients in different video frames is different. 2) We set some high frequency DCT coefficients to "0" before watermarking to improve the robustness of watermark extraction. 3) During a typical video coding the quantization setting for I-frames and B- or P-frames are different—the quantizer step size for I-frames is smaller than those for B- or P-frames (we could also see similar PSNR patterns during multicycle compression without watermarking). 4) Because we watermarked I-frames, B-frames and P-frames independently in compressed domain to maintain the watermarking and authentication robustness, the watermark distortion from I-frames may "spread" to B- or P-frames during reencoding.

The evaluation on file size change before and after signing/watermarking is shown in Fig. 15. We can see that the file sizes did not vary significantly though some videos increased in size while some did not. The video "lobby" is another testing video set up at the local airport for surveillance purposes, shown in Fig. 16.

To evaluate the system's performance robust to video transcoding, we have also developed a video transcoder, which can perform quantizer step change from 1 to 6, format conversion from CIF to QCIF and frame dropping from 30 to 5 fps. The watermarked video sequences are processed by the

Fig. 16. Watermarked video frame (left) versus its attacked video frame (right). The attacked video is generated by removing one person from the watermarked video. The attacked one cannot pass the authentication.

transcoder before verification. The experiments showed that all the transcoded videos whose authentication robustness was predefined, are still authentic. Note that under this setting, the ranges of the achieved average bit rate reduction are 57.8% by requantization, 71.7% by CIF-QCIF conversion, and 66.2% by frame-dropping, respectively (the original average bit rate is about 2.04 Mb/s for five test videos). The most robust test which can still pass the authentication is their combination (i.e., $Q_a = 6$, QCIF and 5 fps). In this case, the transcoded bit rate is about 700 kb/s. More discussions will be given in next subsection.

We also modified the watermarked video to evaluate the security of the scalable authentication system. Fig. 16 shows one watermarked video frame and its attacked version. The attacked video is generated by removing one person from the watermarked video. During verification, the system can successfully detect that the attacked video is unauthentic. We also successfully tested the change of the time info shown on the video.

B. Discussions on System Robustness and Security

Based on the description and testing results given above, we can see that the proposed scheme authenticates the signed video in a configurable way on both system robustness and system security. In other words, the system can achieve different levels of robustness and security by adjusting some parameter settings. The first parameter is the quantization step setting for authentication [10], [15]–[17]. The larger the quantization step we choose, the more robust the system will be. Consequently, it will result in a less secure system (more rooms to be attacked) and a lower signed video quality (more robust watermarking). The second parameter is the rate of frame-dropping. If more frames are dropped, longer hashes have to be embedded into one frame. Because of the limit on watermarking capacity, the fixed length of crypto hash (e.g., 128 bits by MD5) generated from one video frame has to be chopped to a shorter one. For instance, in our simulation, we chopped the crypto hash from 128 to 36 bits to make the system robust to predefined frame-dropping rate (i.e., frame rate change from 30 to 5 fps). Obviously, the security of 36 bits MAC is less than that of 128 bits. The third issue related to system robustness and security is feature selection. A badly selected feature could make it possible for a forger to generate dissimilar images which have the same features [21]. And the authentication system with more selected features will be more secure. As to how to properly set system parameters and select features, we argue that it is application-dependent, i.e., a thorough understanding (e.g., the acceptable manipulations and unacceptable modifications on the video and their strength) of the specific application is the precondition to designing a good authentication scheme whose robustness and security are satisfactory.

The demand on a configurable security scheme comes from scalable media applications such as streaming [22] and MPEG7 universal media access, where content adaptation is the main focus [23]. Such requirement is very important for security in a pervasive environment, which contains many heterogeneous devices with varying computing resources and connected through different channels. Therefore, in such an environment, traditional security solutions, which only provide yes/no answer, cannot work well because different devices and channels may have different required security levels due to their limited computing power or their targeted applications. For example, sending a thumb-size gray-level image to a small portable device demands less security than sending a full-size color image to a desktop computer. This "quality of protection" concept is well aligned with the concept called "quality of service" in network applications and should be more suitable for multimedia-related streaming applications.

VI. Conclusions and Future Work

In this paper, we have proposed a robust and secure authentication solution for video transcoding. We consider three common transcoding approaches (i.e., frame resizing, frame dropping and requantization) as acceptable manipulations for our authentication scheme. Others, such as frame insertion, replacement or partial modifications will be deemed as unauthentic. The proposed scheme achieves an end-to-end authentication, which is independent of specific streaming infrastructure. System robustness and security is balanced in a configurable way that suits for media streaming and adaptation under universal media access. Compressed-domain processing further improves the system computation efficiency and watermarking is employed to reduce the transmission cost of the signed video. The scheme is compliant with PKI except that the video to be transmitted is the watermarked content not the original one.

More technical details are skipped in this paper due to the limit of paper size. We are currently working on the improvements to our implemented prototype. Our future work is to extend the proposed scheme for other practical applications such as online broadcasting. In such cases, we may have to allow the newly joined audiences to authenticate the stream.

References

[1] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996.
[2] A. Vetro, C. Christopoulosa, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, pp. 18–29, Mar. 2003.
[3] H. Sun, W. Kwok, and J. W. Zdeoski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 191–199, Mar. 1996.

[4] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit-streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.
[5] K.-T. Fung, Y.-L. Chan, and W.-C. Siu, "New architecture for dynamic frame-skipping transcoder," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 886–900, Aug. 2002.
[6] R. Gennaro and P. Rohatgi, "How to sign digital stream," *Proc. Crypto'97*, pp. 180–197, 1997.
[7] H. Yu, "Scalable multimedia authentication," in *Proc. 4th IEEE Pacific-Rim Conf. Multimedia*, Singapore, 2003, p. 0886.
[8] J. M. Park, E. K. P. Chong, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *Proc. ACM Trans. Inf. Syst. Security*, vol. 6, no. 2, pp. 258–285, 2003.
[9] Y. Wang, J. Osterman, and Y.-Q. Zhang, *Video Processing and Communications*. Englewood Cliffs, NJ: Prentice Hall, 2002.
[10] Q. Sun, S.-F. Chang, M. Kurato, and M. Suto, "A quantitative semifragile JPEG2000 image authentication system," in *Proc. Int. Conf. Image Process. (ICIP02)*, 2002, pp. II921–II924.
[11] W. Zhu, K. H. Yang, and M. J. Beacken, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs Techn. J.*, pp. 21–29, Jul.–Sep. 1998.
[12] S. Wee, B. Shen, and J. Apostolopoulos, Compressed-domain video processing HP Labs Tech. Report, (Oct. 2002) [Online]. Available: http://www.hpl.hp.com/personal/Susie_Wee/pub-hpl.htm
[13] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, pp. 283–301, May 1998.
[14] M. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Video authentication with self-recovery," in *Proc. SPIE: Security Watermarking Multimedia Contents IV*, Jan. 2002, vol. 4675, no. 58, pp. 532–541.
[15] C.-Y. Lin and S.-F. Chang, "Semi-fragile watermarking for authenticating JPEG visual content," in *Proc. SPIE Security Watermarking Multimedia Contents II, EI'00*, San Jose, CA, Jan. 2000, pp. 140–151.
[16] ——, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 153–168, Feb. 2001.
[17] ——, "Issues and solutions for authenticating MPEG video," in *Proc. SPIE Security Watermarking Multimedia Contents, EI'99*, San Jose, CA, Jan. 1999, pp. 54–65.
[18] Q. Sun, D. He, Z. Zhang, and Q. Tian, "A robust and secure approach to scalable video authentication," in *Proc. Int. Conf. Multimedia Expo (ICME03)*, Baltimore, MD, Jul. 2003, pp. II209–II212.
[19] D. Zheng, J. Zhao, and E. Saddik, "RST-invariant digital image watermarking based on log-polar mapping and phase correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 753–765, Aug. 2003.
[20] D. He, T.-T. Ong, Z. Zhang, and Q. Sun, "A practical watermarking scheme aligned with compressed-domain CIF-to-QCIF video transcoding," in *Proc. 4th IEEE Pacific-Rim Con. Multimedia (PCM03)*, , Singapore, 2003, p. 2043.
[21] C. W. Wu, "On the design of content-based multimedia authentication systems," *IEEE Trans. Multimedia*, vol. 4, no. 3, pp. 385–393, Mar. 2002.
[22] C. S. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile multimedia applications," in *Proc. Int. Conf. Multimedia Expo (ICME03)*, pp. II137–II140.
[23] *MPEG Requirements Group*, MPEG-7 overview, ISO/IEC N4980, 2002.

Qibin Sun received the Ph.D. degree in electrical engineering from the University of Science and Technology of China (USTC), Anhui, China, in 1997.

Since 1996, he has been with the Institute for Infocomm Research, Singapore, where he is responsible for industrial as well as academic research projects in the areas of media security, image and video analysis. He was with Columbia University, New York, during 2000–2001, as a Research Scientist. He is currently leading the Media Understanding Department, Institute for Infocomm Research, Singapore, conducting research and development in media (text, audio, image, video) analysis, retrieval, and security. He is also the Head of Delegates of Singapore in ISO/IEC SC29 WG1(JPEG).

Dr. Sun actively participates in professional activities such as IEEE ICME, IEEE ISCAS, IEEE ICASSP, and ACM MM, etc. He serves as a member of the Editorial Board in IEEE *Multimedia Magazine* and *LNCS Transactions on Data Hiding and Multimedia Security*, and is an the Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Dajun He received the B.S. degree from Tsinghua University, Tsinghua, China, in 1991, the M.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1994, and the Ph.D. degree from the National University of Singapore, Singapore, in 2005.

From 1994 to 1995, he was a Lecturer with Shanghai Jiaotong University, where he developed the first HDTV simulation system in China. From 1996 to 2001, he was a Senior Engineer with AIWA Singapore, developing audio and visual consumer products. From 2001 to 2005, he was a Scientist with the Institute for Infocomm Research, Singapore. Currently, he is a Deputy Director of Engineering with Shanghai Zhangjiang (Group) Co., Ltd., China. His main research interests include media security and image/video processing.

Qi Tian (S'83–M'86–SM'90) received the B.S. and M.S. degrees from the Tsinghua University, Beijing, China, and the Ph.D. degree from the University of South Carolina, Columbia, respectively, all in electrical and computer engineering.

He is currently a Principal Scientist at the Institute for Infocomm Research, Singapore. His main research interests are image/video/audio analysis, multimedia content indexing and retrieval, computer vision, pattern recognition, and machine learning. He joined the Institute of System Science, National University of Singapore, in 1992 as a member of research staff, and subsequently became the Program Director for the Media Engineering Program at the Kent Ridge Digital Laboratories, then Laboratories for Information Technology, Singapore (2001–2002). He has published over 110 papers in peer reviewed international journals and conferences.

Dr. Tian served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1997–2004), and served as chairs and members of technical committees of international conferences such as the IEEE Pacific-Rim Conference on Multimedia (PCM), the IEEE International Conference on Multimedia and Expo (ICME), ACM-MM, and Multimedia Modeling (MMM).