Open access • Proceedings Article • DOI:10.1117/12.344703

Issues and solutions for authenticating MPEG video — Source link

Ching-Yung Lin, Shih-Fu Chang

Institutions: Columbia University

Published on: 09 Apr 1999 - electronic imaging (International Society for Optics and Photonics)

Topics: Video post-processing, Video processing, Transcoding, Authentication and Digital watermarking

Issues and Solutions for Authenticating MPEG Video

Ching-Yung Lin and Shih-Fu Chang

Department of Electrical Engineering &
New Media Technology Center
Columbia University, New York, NY 10027, USA

ABSTRACT

Video authentication techniques are used to prove the originality of received video content and to detect malicious tampering. Existing authentication techniques protect every single bit of the video content and do not allow any form of manipulation. In real applications, this may not be practical. In several situations, compressed videos need to be further processed to accommodate various application requirements. Examples include bitrate scaling, transcoding, and frame rate conversion. The concept of asking each intermediate processing stage to add authentication codes is flawed in practical cases. In this paper, we extend our prior work on JPEG-surviving image authentication techniques to video. We first discuss issues of authenticating MPEG videos under various transcoding situations, including dynamic rate shaping, requantization, frame type conversion, and re-encoding. Different situations pose different technical challenges in developing robust authentication techniques. In the second part of this paper, we propose a robust video authentication system which accepts some MPEG transcoding processes but is able to detect malicious manipulations. It is based on unique invariant properties of the transcoding processes. Digital signature techniques as well as public key methods are used in our robust video authentication system.

Keywords: video authentication, content verification, tampering detection, MPEG, digital signature, watermark.

1 Introduction

Video authentication, a multimedia security enhancement technique, protects the recipients against malicious forgery. Today, editing or modifying the content of a digital video can be done efficiently and seamlessly, and the credibility of digital data decreases seriously. To ensure trustworthiness, authentication techniques are needed for verifying the originality of video content and preventing forgery. In this paper, we address the importance of Multimedia Content Authentication (MCA), and focus on the issues and solutions of digital video authentication.

1.1 Multimedia Authentication

Authenticity, by definition, means something "as being in accordance with fact, as being true in substance", or "as being what it professes in origin or authorship, as being genuine[1]." A third definition of the authenticity is to prove that something is "actually coming from the alleged source or origin[2]." For instance, in the courtroom, insurance company, hospital, newspaper, magazine, or television news, when we watch/hear a clip of multimedia data, we hope to know whether the image/video/audio is *real*. For electronic commerce, once a buyer purchases multimedia data from the internet, she needs to know whether it comes *from the alleged producer* and she must be assured that *no one has tampered with the content*. From the definition of authenticity and the above examples, we feel that the credibility of multimedia data is expected for the purpose of being *electronic evidence* or a *certified product*. The different requirements for authenticity affect the methodologies and design in

the practical application systems[3].

Multimedia data may be a representation of fact, a production of original sources (artifacts, documents or works of art), or a re-interpretation of sources (that are either in the digital format or in other formats). In contrast with traditional original sources whose authenticity can be established from many physical clues, multimedia data is a combination of abstract bits which can only be authenticated by non-physical clues. One possibility, which can be called as *blind authentication*, is to examine the characteristics of content for authorship inference and the continuity of content for forgery detection. This method is widely used in traditional authentication applications, but it is still under development for multimedia authentication. A practical solution is the *digital signature* method introduced by Diffie and Hellman in 1976[4]. The digital signature of the signer to the data shall depend on the content of data on some secret information only known to the signer[5]. Therefore, the digital signature can not be forged, and the authenticator can verify a received multimedia data by examining whether its contents match the information conveyed in the digital signature. In other words, digital signature can be used to verify the data integrity which is endorsed by the signer. This way, the authentication mechanism is based on trusting the signer.

It is more difficult for a machine signer to lie on or deny the digital signature than a human signer. An approach to the solution of verifying whether a clip of multimedia data is *real* is the trustworthy camera proposed by Friedman in 1993[6]. Based on the encryption chip in the camera, it endorses its captured multimedia data. Because a rigid camera cannot lie on the digital signature, it can provide some credibility of the reality to the captured multimedia data.

We should notice that no matter how the authentication algorithm can be, the trustworthiness of the endorser will be always an important concern. In the traditional research of message authentication, the endorser is usually the one who generates and distributes the message. However, multimedia data are usually distributed and re-interpreted by many interim entities. Then, it will be important to reduce the necessary number of endorsers. That can be achieved by the robust digital signature method that we have proposed[7, 8, 10]. The robust digital signature is used for the purpose of multimedia content authentication which we will discuss in the next section.

1.2 Multimedia Authentication Objectives: Complete Verification v.s. Content Verification

Based on the objectives of authentication, we can categorize an authentication system as *complete verification* and *content verification*. Techniques for complete verification consider multimedia data as untouchable messages such that the data for authentication have to be *exactly* the same as the original one. Previous works of the message authentication in the cryptography field were all in this category.

Content Verification is a characteristic of multimedia data authentication. Because the meaning of multimedia data is based on their content instead of the bitstreams, in some applications, manipulations on the bitstreams without changing the meaning of content are considered as acceptable. Compression is an example. Today, most digital multimedia data are stored or distributed in compressed forms. To satisfy the various needs of broadcasting, storage and transmission, some transcoding of compressed digital videos may be required. For instance, digital video clips are usually shot and stored in the compressed format with a pre-determined bitrate, but distributed with a different final bitrate. Transcoding processes change the pixel values of the digital video but not its content. Therefore, videos that are transcoded from the original should be considered as authentic.

If the multimedia data is served as a product in the electronic commerce, the seller can confine the use of the product *in a priori* and only endorse it by complete authentication. On the other hand, in order to extend the use of the product, the seller can also endorse it by content authentication. However, if we need to authenticate a clip of multimedia data which may serve as an electronic evidence *in a posteriori* and may have been manipulated by acceptable methods, then only the content authentication can be used in this scenario.

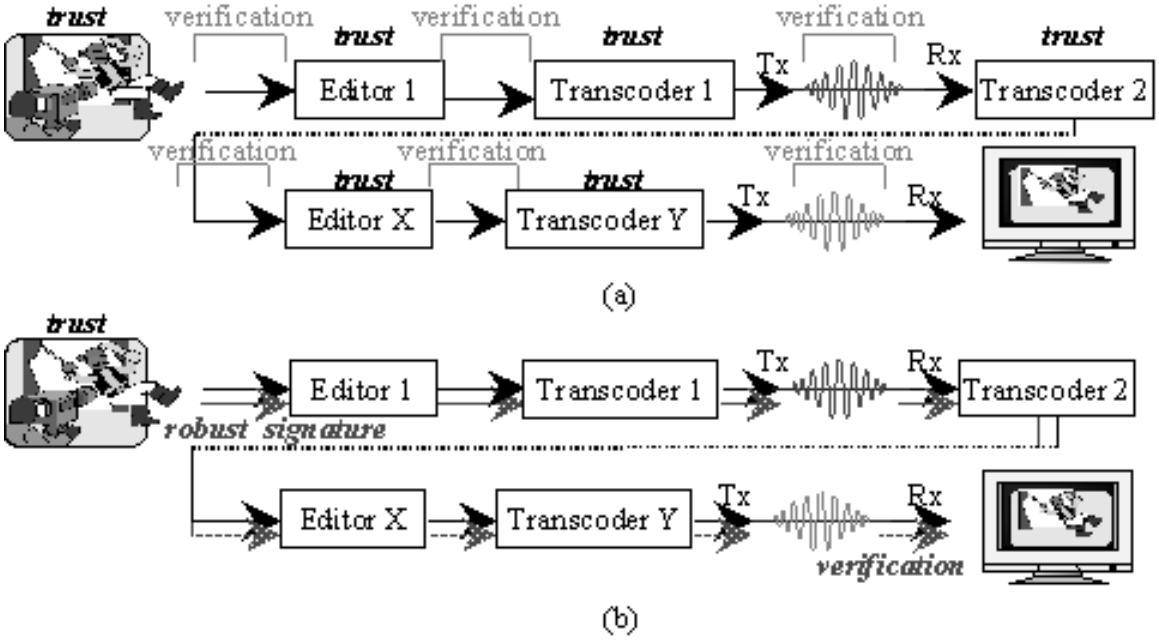

Figure 1: (a) Complete Verification: multimedia data have to be examined in each transmission, and each intermediate stage must be trustworthy; (b) Content Verification: multimedia data are endorsed by the producer and verified only in the last stage.

Figure 1 shows the benefit of the Multimedia Content Authentication (MCA). It represents the complete process of multimedia data, from being produced to being consumed. With complete verification, we have to verify the data at every transmission stage and trust all the interim entities. However, with content verification, we can transmit the robust signature with the data and only verify it at the last stage. Therefore, we do not need to verify the data at each stage and risk the trustworthiness of the intermediate people. This enhances the authenticity of the data. Moreover, with complete verification methods, the data cannot get the endorsement from the producer unless there are no intermediate stages. However, by the robust signature, the originality of the multimedia data can be endorsed by the producer. As in Figure 1, if the producer is a trustworthy camera, it can somehow provide credibility of reality to the data, *i.e.*, proving that the multimedia data are "real." This is especially useful for those multimedia data that are used as electronic evidence.

1.3 Multimedia Authentication Sources: Raw Data v.s. Compressed Data

Multimedia compression standards have been designed and widely adopted by various applications such as JPEG in the WWW, MPEG-1 in the VCD, MPEG-2 format in the HDTV, and H.261 and H.263 in the video phone. The source of an multimedia authentication system may be raw data or compressed data. In practical applications, the raw format of multimedia data may not be available. For instance, a scanner generates temporary raw images but only saves them in their compressed format; a digital camera which captures image/video produces compressed files only without generating any raw format data. Therefore, an authentication system which can only authenticate raw data will have limited uses in practice. Examples of these special cases include: (1) non-standard data such as 3D objects, and (2) medical images which do not tolerate lossy compression.

Researcher	*Method*		*Objective*		*Source*	
	Digital Sig.	Watermark	Complete Ver.	Content Ver.	Raw Data	Compressed Data
Friedman[6]	X		X		X	X
Van Schyndel *et. al.*[11]		X	X		X	
Walton[12]		X	X		X	
Wolfgang and Delp[14]		X	X	X	X	
Zhu *et. al.*[13]		X		X	X	
Schneider and Chang[7]	X			X	X	X
Yeung and Mintzer[15]		X	X		X	
Lin and Chang[8, 9]	X			X	X	X

Table 1: Previous Research Work

1.4 Multimedia Authentication Methods: Watermarking v.s. Digital Signature

In addition to the digital signature method, there is an alternative for multimedia authentication: *watermarking*. Since the meaning of multimedia data is based on its content, we can modify the multimedia bitstream to embed some codes, *i.e.*, watermarks, without changing the meaning of the content. The embedded watermark may represent either a specific digital producer identification label (PIL) or some content-based codes generated by applying a specific rule. Because the watermarks are embedded in the data content, once the data is manipulated, these watermarks will also be modified such that the authenticator can examine them to verify the integrity of the data.

For complete verification of uncompressed raw multimedia data, watermarking may work better than digital signature methods because:

- the watermarks are always integrated with the data such that the authenticator can examine them conveniently, and
- there are many spaces in the multimedia data to embed the watermarks without degrading the quality too much (or even make the watermarks invisible).

Previous works in [11, 12, 15] have shown the effectiveness of the watermarking methods for this type of application.

However, there is no advantage to use the watermarking method in a compressed multimedia data for complete verification. Compression standards,*e.g.*, MPEG or JPEG, have user-defined sections where digital signature can be placed. Because multimedia data are stored or distributed in the file format instead of pixel values, therefore, the digital signature can be considered as being "embedded" in the data. Once the multimedia data is modified, the user-defined section of the original data is usually discarded by the editing software. Even if the digital signature can be reserved by the software, we can easily detect the modification, since the hash values of the modified data will not be the same as the original. Moreover, because there is less space for compressed multimedia to hide watermarks, if we do not want to sacrifice too much visual quality on the multimedia data, there may not be enough information bits to protect the data.

For content verification, a watermarking method that can reliably distinguish compression from other manipulations still has not been found. The watermarks are either too fragile for compression or too flexible for manipulations. The performance of an authenticator should be simultaneously evaluated by two parameters: the probability of false alarm and the probability of missing manipulations. Fragile watermarks, which have low probability of miss, usually fail to survive compressions such that their probability of false alarm is very high. Previous researchers have attempted to modify the fragile watermark to make it more robust with compression[13, 14]. However, such modifications failed to distinguish compression and tampering. When they lower the probability of false alarm, the probability of miss in their systems increases significantly. On the other hand, robust watermarks, previously used for copyright protection, are robust with most manipulations, but are usually too robust to detect malicious manipulations. For these watermarks, the probability of missing manipulation is usually too high.

	Situation 1	Situation 2	Situation 3	Situation 4	Situation 5
DCT (residual) coefficients	X (drop some coefficients)	X (requantization)		X	
Motion Vectors	X	X	X	X	
Picture Type (I,P,B)	X	X	X	X (inconsistent in boundary)	

Table 2: Consistent Properties of Transcoding and Editing Processing Situations

Digital signatures can be saved in two different ways. If the header of the compressed source data remains intact through all processing stages, then the digital signature can be saved in the header. Otherwise, it can be saved as an independent file. Anyone who needs to authenticate the received multimedia data has to request the source to provide the signature. This may be inconvenient in some cases and considered as a drawback. But, since the digital signatures remain unchanged when the pixel values of the images/videos are changed, they provide a better prospect for achieving robustness. Currently, only robust digital signature methods are proved to be useful for content verification[9]. Because our techniques were based on the characteristics of DCT-based compression standards, they can exactly distinguish compression from other manipulations.

In Table 1, we compare different existing authentication techniques based on the type of method used, the objective, and the source data being authenticated.

In the following sections, we will focus on the issues and solutions of authenticating MPEG video. To extend our previous image authentication techniques, two important issues have to be noted: (1) transcoding and editing processes and (2) size of the digital signature. Since digital videos are seldom recorded in their raw format, we will consider the sources for authentication as being all in either the MPEG-1 or MPEG-2 format.

2 MPEG Video Authentication

To design a system for the content authentication of compressed video, we have to know the types of possible acceptable manipulations that may be applied to the video. In general, five acceptable transcoding or editing processing situations may be applied to the compressed video:

1. Dynamic Rate Shaping[19, 20]: A real-time rate-control scheme in the compressed domain. This technique sets dynamic control points to drop the high-frequency DCT coefficients on each 8×8 block in a macroblock. Motion vectors are not changed.

2. Rate Control without Drift Error Correction[21, 22]: This technique is also applied in the compressed domain. DCT coefficients are re-quantized to satisfy different bit-rate constraints. Motion vectors are not changed.

3. Rate Control with Drift Error Correction[17]: This technique improves the video quality after the requantization of DCT coefficients, but it needs more computation. DCT coefficients of the residue of intercoded blocks are modified to satisfy the change of the re-quantized intracoded blocks. Motion vectors are not changed in this case.

4. Editing with Mostly Consistent Picture Types[17, 18, 24]: The picture types (I, P and B) are kept unchanged in each editing generation. It may be used in creating a new sequence by cutting and pasting several video segments. The GOP (Group of Pictures) boundaries in each segment are not changed except those near the cut positions. Pixel values may be changed for video quality improvement such as intensity change, filtering, *etc.*.

5. Editing or Transcoding with Inconsistent Picture Types[17]: In some processes, the compressed videos are transformed to the uncompressed bitstreams which are then edited and re-encoded. The GOP structures

and the motion vectors may change in this case. This kind of process includes format transmission between different compression standards and picture type conversion.

The first three processes are used for bitrate changes. They are all operated in the compressed domain. In other words, the structure of the MPEG *Program Streams* do not change. From Table 2, we can know that after these transcoding processes, the motion vectors and the picture types are preserved. The only change is on either the DCT coefficients of the *intra macroblocks* or the DCT residual coefficients of the *non-intra macroblocks*.

In studios, cutting and pasting several MPEG video segments to create a new video sequence is very common. It can be done with two different methods, Processing Situation 4 and Processing Situation 5. Their difference is basically whether the GOP structure is preserved through the editing process. In Situation 4, there are two kinds of GOP in the generated video sequence: original GOPs and created GOPs. An original GOP comes from an original video sequence with its structure intact. The created GOPs are generated from the boundary pictures of the original video sequence(s). There may be no created GOPs if the video sequence is not allowed to be cut inside a GOP. In practice, the number of created GOPs is much smaller than that of original GOPs (a typical GOP is about 0.5 second). For this situation, we focus on authentication of the original GOPs.

Video authentication signatures can be generated for different situations. We can find that for Situation 1-4, the GOP structure is not modified after transcoding or editing processes. Therefore, we can generate a robust digital signature which can survive these acceptable manipulations. We called this a Type I robust digital signature, which will be discussed in Section 3.1.

For Situation 5, because the GOP structure has been destroyed, only the pixel values of pictures will be preserved. Therefore, the video sequence is like a set of image frames, which can be authenticated by the image authentication that we proposed in [9]. We call this a Type II robust digital signature. The generation method is shown in Section 3.2.

2.1 Syntax of a MEPG Video Sequence

In the MPEG standard, each Video Sequence is composed of several sequential Group of Pictures (GOP). A GOP is an independent unit which includes several Pictures. In MPEG-1, each frame is a Picture. In MPEG-2, a Picture can be either a Field-Picture or a Frame-Picture. There are several Slices in a Picture. A Slice is a string of consecutive MacroBlocks (MBs) of arbitrary length running from left to right across the picture. The MB is the 16×16 motion compensation unit which includes several 8×8 Blocks. (An MB includes 6 blocks with the 4:2:0 chroma format, 8 blocks with the 4:2:2 chroma format, or 12 blocks with the 4:4:4 chroma format.) Each block is either Intra-coded or Non-Intra-coded. In MPEG, as with JPEG, Intra-coded blocks have their DC coefficients coded differently with respect to the previous block of the same YCbCr type, unless the previous block is Non-Intra, belongs to a skipped macroblock (MB), or belongs to another Slice[25]. The AC coefficients of each block in a macroblock is quantized by the *quantization_step_size* which is given by

$$(\kappa \cdot Q[m][n])/(8 \cdot v), \quad m, n = 0, 1, ..7, \;\; m + n \neq 0, \tag{1}$$

where κ is the *quantizer_scale* and Q is the quantization matrix which is either the *Intra Qmatrix* for Intra blocks or the *NonIntra Qmatrix* for Non-Intra blocks. Both blocks may be defined in the *VideoSequence* Header if they are different from the default values. (In the 4:2:0 format, the luminance and chrominance Q-matrices are always the same. But they can be different in the 4:2:2 or 4:4:4 formats.) The parameter v is equal to 1 for MPEG-1 video sequences, or 2 for MPEG-2 video sequences. The *quantizer_scale*, κ, is set for a Slice or a MB.

3 Robust Digital Signature

3.1 Robust Digital Signature: Type I

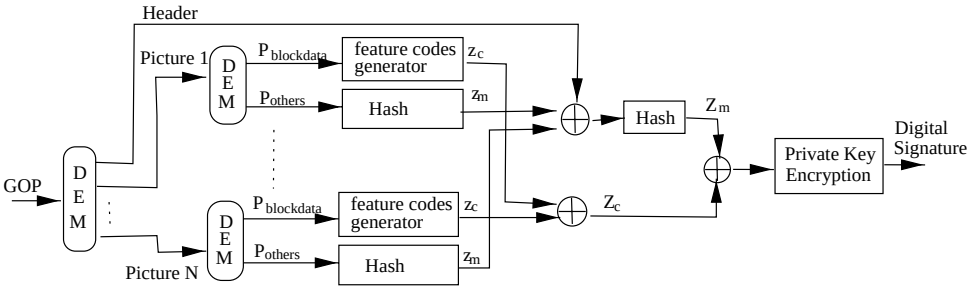

Figure 2: Robust Digital Signature : Type I

In [8, 9], we have shown that the relationship between a coefficient pair, *i.e.*, two DCT coefficients of the same coordinate position, in any two 8×8 blocks of an image should remain the same or become equal after the re-quantization processes, if the same *quantization_step_sizes* are applied on the blocks. We have also shown that the change of the difference value of a coefficient pair after re-quantization should be bounded in a range specified by the *quantization_step_sizes*, which can be different, of the blocks. Therefore, we can arrange all the blocks in an image to form block pairs, and generate some codes to represent the relationship status of coefficients in selected coordinate positions. The generated codes are then encrypted by public key method to form a digital signature.

To generate a robust digital signature for Processing Situations 1-4, we can use the quantized (*intra* or *non-intra*) DCT coefficients of the luminance and chrominance matrices in each macroblock to form comparison pairs. Since the κ value as well as the *quantization_step_size* is always the same in all blocks of a macroblock, the relative relationships of the coefficients at the corresponding positions of blocks are invariant during transcoding. Therefore, similar to the signature generation process of images, we can use them to generate feature codes. First, the feature codes Z_c of a macroblock can be written as,

$$z_c = VLC(\bigcup_{\mathbf{p}} \bigcup_{\mathbf{b}} sgn[\, \mathbf{f_p(b)} - \mathbf{f}_{W(\mathbf{p})}(\mathbf{b}) \,] \,) \tag{2}$$

where

- $\mathbf{f}$ represents the quantized DCT coefficients in the compressed video sequence. They should be extracted from the bitstream and decoded with Variable Length Decoding (VLD).
- $\mathbf{p}$ is the set of the selected blocks in the macroblock, and W is the mapping function which maps each block at $\mathbf{p}$ to its corresponding block for forming a block pair. For instance, in a 4:2:0 format, if we label the 4 luminance blocks and the two chrominance as Block 1-6, then we can choose $\mathbf{p}$ as $\{1, 3, 5\}$ and a set $\mathbf{q} = W(\mathbf{p}) = \{2, 4, 6\}$ which forms three block pairs of Block $\{1, 2\}$, $\{3, 4\}$ and $\{5, 6\}$. For a macroblock of $\wp$ blocks, there will be $\wp!$ combinations.
- $\mathbf{b}$ is the set of the selected DCT coefficient positions. They are represents by the zig-zag order or alternative scan order whichever is used in the Video Sequence. For instance, if we choose to compare the DC values and the 1 - 5 AC coefficients in a block pair, then the $\mathbf{b}$ will be $\{1, 2, 3, 4, 5, 6\}$. The selection of $\mathbf{b}$ can vary for different block pairs.
- the sign function is defined as (1) $sgn(f) = 1,\ if\ f > 0$, (2) $sgn(f) = 0,\ if\ f = 0$, and (3) $sgn(f) = -1,\ if\ f < 0$.

It should be noted that here we use the sign function to represent the difference values because there are lots of zeros in the DCT coefficients of the compressed video sequence. From the viewpoints of information, we should distinguish it from the other two situations, *i.e.*, positive and negative. This is different from what we have done for the images[8, 9]. Because there are lots of zeros in the coefficient comparison results, the VLC method can be applied to reduce the length of the feature codes.

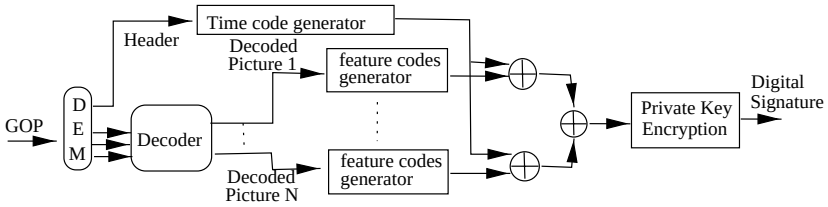

Figure 3: Robust Digital Signature : Type 2

In addition to the protection of DCT coefficients, we need to protect other information including the motion vectors and control codes as well. This can be done by adding the hash values of the remnant bitstreams in the video sequence to the feature codes. At the first step, assume a Picture P, P includes P_{block_data} and P_{others}, where P_{block_data} includes the codes of DCT coefficients, the *quantizer_scale* in the Slice or MB header, and their control codes. P_{others} includes all other codes in P. Then, we can get the hash values as,

$$z_m = Hash(P_{others}) \tag{3}$$

where z_m is used for protecting other information of a Picture.

Because the GOP is the basic independent unit of a Video Sequence in the MPEG bitstream, we can encrypt the feature codes and the hash values of each pictures in a GOP to form a digital signature, *i.e.*,

$$DS = Private\ Key\ Encrypted(Z_c, Z_m) \tag{4}$$

where $Z_c = \bigcup_{Pictures} VLC(\bigcup_{MBs} z_c)$ is a combination of the feature codes z_c of all the macroblocks in the GOP, and

$$Z_m = Hash(GOP\_Header, z_{m,1}, z_{m,2}, ..., z_{m,N}) \tag{5}$$

where N represents the total number of Pictures in a GOP. Eq.(5) indicates that, instead of using the combination of the hash values of each picture, the length of Z_m can be further shortened by hashing the combination values, because all these information are fixed during the transcoding processes. Since GOP_Header includes the *time_code* which refers to the first picture to be displayed after the GOP header that has a *temporal_reference* of zero, it is important to include it to the digital signature for preventing temporal perturbation of GOPs. This digital signature, DS, can be placed in the *user_data* area of the GOP header. (In MPEG standards, *user_data* can be embedded in the Sequence Header, the GOP Header, or the Picture Header.)

3.2 Robust Digital Signature: Type II

The second type of robust digital signature is designed for surviving processes in Situation 5. Since the GOP structure, motion vectors, or DCT residual coefficients may change in this situation, the only consistent property is the pixel values of pictures. Therefore, we have to generate digital signature based on the pixel values of each picture. By using a similar authentication method for images, we can generate the digital signature picture by picture. The generation method is as follows:

1. Reconstruct the pixel values of the picture of any kind of picture type (I, P, B).
2. Generate feature codes by using exactly the same procedure as we proposed in [9], *i.e.*, dividing the image into 8×8 blocks, forming block pairs, comparing the DCT coefficients at the block pair, using one bit to represent each comparison.
3. Add time codes of each picture to the feature codes.
4. Using the Private Key Encryption to form the digital signature.

A diagram of the generating this type of robust digital signature is shown on Figure 3.

4 Authenticator

4.1 Authenticating Video Sequence after Transcoding (Situations 1-3)

The authenticator can be implemented as an augmentation of the general decoder. In the authenticator, the digital signature is extracted from the GOP header and decrypted to get the feature codes and the hash values. For examining the authenticity of a GOP in the video sequence, similar to the processes of signature generation, each picture in the GOP is divided into two parts: P_{block_data} and P_{others}. We then authenticate these two parts separately. To authenticate the hash values, we can get the $\hat{Z}_m$ of the GOP by using the same hash function(s) in the Eq. (3) and Eq. (5). Since this part of information is intact during the transcoding processes, $\hat{Z}_m$ is expected to be equal to Z_m. Otherwise, this GOP must have been modified by some other processes.

To authenticate the feature codes of GOP, the authenticator must first apply the VLC decoding to the feature codes to obtain the signs of the relationship of selected coefficients in each block pair. By applying a similar procedure of the authenticator we proposed on [8, 9], we can authenticate whether the DCT coefficients have been maliciously modified because:

- in Situation 1, some DCT high frequency coefficients in a block may be dropped and set to zero. Referring to the Theorem 1 in [9], if two DCT coefficients are both equal to zero after transcoding, the authenticator considers them as authentic. Because the lower frequency coefficients are preserved during transcoding, their relationships will be exactly the same as the original.

- in Situation 2, the DCT coefficients may be requantized to satisfy some bitrate constraints. Since all the DCT coefficients at the same position of the blocks in a MB are always quantized by the same *quantization_step_size*, according to the same theorem in [9], the possible changes of the sign values of the difference of a coefficient pair are: "positive to positive," "positive to zero," "zero to zero," "negative to negative," and "negative to zero." If we find the relationships of the coefficients do not satisfy this rule, we can claim that the video sequence has been modified by other manipulations.

- in Situation 3, the DCT coefficients of the intra blocks may be requantized. Also, the DCT residue coefficients of the non-intra blocks may be changed to compensate the quantization error introduced by the requantization of their reference blocks, and then be requantized again. To authenticate these blocks, we can introduce some tolerance bound to the authenticator. If we define $\Delta\mathbf{f}_{p,q}(b) = \mathbf{f}_p(b) - \mathbf{f}_{W(p)}(b)$, which is the difference of the coefficients at the position b in the block pair $(p, W(p))$ of the original video, and $\Delta\hat{\mathbf{f}}_{p,q}(b) = \hat{\mathbf{f}}_p(b) - \hat{\mathbf{f}}_{W(p)}(b)$, which is from the examined video. Then, the following property has to be satisfied,

$$if\ \Delta\mathbf{f}_{p,q}(b) > 0, \quad then \quad \Delta\hat{\mathbf{f}}_{p,q}(b) \geq -\tau, \tag{6}$$

$$else\ if\ \Delta\mathbf{f}_{p,q}(b) = 0, \quad then \quad \tau \geq \Delta\hat{\mathbf{f}}_{p,q}(b) \geq -\tau, \tag{7}$$

$$else\ if\ \Delta\mathbf{f}_{p,q}(b) < 0, \quad then \quad \Delta\hat{\mathbf{f}}_{p,q}(b) \leq \tau. \tag{8}$$

where

$$\tau = \begin{cases} 0, & intrablock, \\ 1 + \sum_{\mathbf{i}} \frac{\hat{\kappa}_{ref_i} \cdot Q_{ref_i}(b)}{\hat{\kappa} \cdot Q_{nonintra}(b)}, & nonintrablock \end{cases} \tag{9}$$

In Eq.(9), $\hat{\kappa}$ is the *quantizer_scale* of the nonintra blocks p and q in the examined video sequence. The set $\mathbf{i}$ represents the number of reference blocks, *e.g.*, $\mathbf{i} = \{1\}$ for a non-intra block in the first P picture of GOP, or $\mathbf{i} = \{1, 2\}$ for a non-intra block in the second P picture of GOP. The parameters $\hat{\kappa}_{ref_i}$ and Q_{ref_i} are the *quantizer_scale* and the quantization matrix of the i-th reference block, respectively. (For a bi-directional predicted non-intra block, we have to use the average of the $\hat{\kappa}_{ref} \cdot Q_{ref}$ from its two reference blocks.) The proof of Eq.(9) is shown in [26]. Similar to the previous situations, the authenticator can examine the coefficients by Eq.(6)-(8). If they are not satisfied, we know that the video sequence must have been modified by unacceptable manipulations.

In addition to the manipulations within the GOPs, an attacker may perturb the temporal order of GOPs to change the meaning of video sequence. This manipulation can be detected by examining the time codes on the GOP Header that are protected in the digital signature. Changes of temporal order of pictures in a GOP can be detected because both feature codes and hash values of the digital signature are generated in the order of pictures.

4.2 Authenticating Video Sequence after Editing (Situations 4 and 5)

The Type I robust digital signature is used in the Situation 4. In this situation, there are two kinds of GOP in the generated video sequence: original GOPs and created GOPs. For an original GOP that comes from an original video sequence with its structure intact, it has its independent digital signature which can be examined by the same authentication method described earlier. The created GOPs are generated from the boundary pictures of the segments of the original video sequence(s). There may be no created GOPs if we restrict the video sequence cannot be cut inside a GOP. This means splicing can only be performed to a resolution of about half a second[23]. If this restriction can not be satisfied, in a created GOP, type conversions may be applied on some pictures[24]. In an compressed video editor, if the digital signature of the corresponding source GOP is copied to the header of the created GOP, then those pictures without type conversions as well as all the intracoded macroblocks can be examined. The authenticator cannot examine those pictures with type conversions. Otherwise, if the digital signature is not copied to the created GOP, there is no clue for examining the authenticity. In general, we can neglect these boundary pictures and show that they are not examined. It addition to authenticating video sequences after cutting and pasting in the temporal segments, some other editing processes such as intensity enhancement, cropping, scaling, filtering, *etc.* may be applied in the video sequences. The robustness of our proposed digital signature towards these manipulations has been shown in [10].

For Situation 5 (video cut & paste or transcoding), all pixel values in each picture may change. However, the changes are like noises and are usually small such that they do not change the meaning of video content. As we have discussed in [10], small noise-like changes in the spatial domain result in small changes in the DCT domain, too. Therefore, large changes in the DCT domain can be assumed to be from malicious manipulations. We can authenticate each picture by some pre-determined tolerance values, τ, in the authenticator. Applying Eq.(6)-(8), if all the coefficient pairs satisfy the equations, we can claim that authenticity of the examined video sequences.

Because there is no exact tolerance bound for changes caused by transcoding or editing of Situation 5, the authenticator can only indicate some areas of a picture "may have been" maliciously manipulated. This is done by observing the authentication result of the picture with different tolerance values. For instance, if $\tau = 0$, we may find the authenticator considers a large percentage of blocks in the picture as being manipulated. However, as τ increases, we can observe that most false-alarms will disappear and only areas that are actually maliciously manipulated are detected by the authenticator.

The time codes that are included in the digital signature can be used to detect changes in the temporal order and indicate the pixel values in the picture of the specific time. Since the video sequence is authenticated picture by picture, its authenticity can still be examined even if it was re-encoded with different temporal resolution.

5 Experimental Results and Discussion

Several video sequences have been tested with our proposed algorithms by using two different digital signatures. Our purpose is to evaluate the probability of missing a malicious manipulation by minimizing the probability of falsely reporting a manipulation. Through 20 more practical experiments on a video sequence "train" (*e.g.*, transcoding in different rates, editing with cut and paste, object substitutions, *etc.*), we found that in Situations 1,2, and 4, there was no false alarm with tolerance values, $\tau = 0$, and in Situation 3 and 5, there was no false alarm with $\tau = 2$. With those settings, the authenticator can detect all object substitution manipulations.

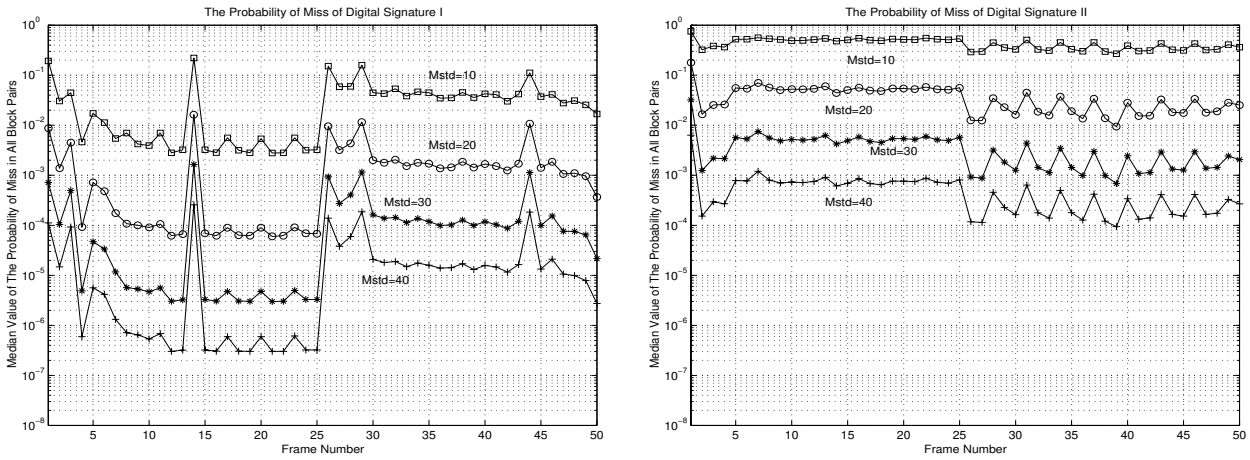

Figure 4: The Probability of Miss of Digital Signature I and II

Further system performance analysis can be done by estimating the probability of miss. The details of the statistical analysis are shown in [26]. In Fig.4, we show an example of the probability of miss of the video sequence "train", which includes 50 frames. From Fig.4(a), we can observe them with four different manipulation levels in terms of the standanrd deviation of Gaussian distributed manipulation changes in the blocks. In this example, we use six coefficients compared in a block pair. For the typical level of manipulation in the range of 30 - 40[9], we can see the probabilities of miss of frames are within the range of 10^{-7} to 10^{-3}. Those are all quite small. By examining the original video sequence, I frames or P frames with more intra blocks have larger probability of miss. That comes from the fact that intra blocks have more nonzero coefficients that are more insensitivie to manipulations. From Fig.4(b), we can find that the probabilities of miss of Digital Signature Type II are larger than those in Fig.4(a). The probability of miss is in the range of 10^{-4} to 10^{-2}. There are two reasons for this phenomeon. The first one is because all the blocks are decompressed and all of them are considered as the intra blocks. The second reason is the use of a larger tolerance value ($\tau = 2$), which reduces the probability of false alarm but also increases the probability of miss.

Using the above practical simulations and mathematical analyses, we have examined the effectiveness of the proposed digital signature algorithms. Our technique can distinguish compression from malicious manipulations. It solves the blind trustworthy problem of interim entities and makes video content authentication feasible. Currently, we are investigating issues in the MPEG audio content authentication for a complete multimedia content authentication system.

References

[1] The Oxford English Dictionary, 2nd Ed., *Oxford Univ.*, pp. 795-796, 1989.

[2] The Webster's New 20th Century Dictionary.

[3] D. Bearman, and J. Trant, "Authenticity of Digital Resources: Towards a Statement of Requirements in the Research Process," *D-Lib Magazine*, June 1998.

[4] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. on Information Theory*, Vol. 22, No. 6, pp.644-654, Nov 1976.

[5] P. Wohlmacher, "Requirements and Mechanisms of IT-Security Including Aspects of Multimedia Security," *Multimedia and Security Workshop at ACM Multimedia 98*, Bristol, UK, Sep 1998.

[6] G. L. Friedman, "The Trustworthy Digital Camera: Restoring Credibility to the Photographic image," *IEEE Trans. on Consumer Electronics*, Vol.39, No.4, pp.905-910, Nov 1993.

[7] M. Schneider and S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication," *IEEE International Conf. on Image Processing*, Laussane, Switzerland, Oct 1996.

[8] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Surviving JPEG Lossy Compression," *SPIE Storage and Retrieval of Image/Video Databases*, San Jose, Jan 1998.

[9] C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation," *CU/CTR Technical Report 486-97-19*, Dec 1997.

[10] C.-Y. Lin and S.-F. Chang, "Generating Robust Digital Signature for Image/Video Authentication," *Multimedia and Security Workshop at ACM Multimedia 98*, Bristol, UK, Sep 1998.

[11] R. G. van Schyndel, A. Z. Trikel, and C. F. Osborne, "A Digital Watermark," *IEEE International Conf. on Image Processing*, Austin, Texas, Nov 1994.

[12] S. Walton, "Image Authentication for a Slippery New Age," *Dr. Dobb's Journal*, pp. 18-26, April 1995.

[13] B. Zhu, M. D. Swanson, and A. H. Tewfik, "Transparent Robust Authentication and Distortion Measurement Technique for Images," *The 7th IEEE Digital Signal Processing Workshop*, pp. 45-48, Sep 1996.

[14] R. B. Wolfgang and E. J. Delp, "A Watermark for Digital Images", *IEEE International Conf. on Image Processing*, Laussane, Switzerland, Oct 1996.

[15] M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE International Conf. on Image Processing*, Santa Barbara, Oct 1997.

[16] R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," *Proceeding of CRYPTO 97*, Santa Barbara, CA, USA, Aug 1997.

[17] P. N. Tudor and O. H. Werner, "Real-Time Transcoding of MPEG-2 Video Bit Streams," *International Broadcasting Convention (IBC 97)*, Amsterdam, Netherlands, pp. 286-301, Sep 1997.

[18] O. H. Werner, "Generic Quantiser for Transcoding of Hybrid Video," *Proceedings of the 1997 Picture Coding Symposium*, Berlin, Germany, Sep 1997.

[19] A. Eleftheriadis and D. Anastassiou, "Constrained and General Dynamic Rate Shaping of Compressed Digital Video," *Proceedings of the 2nd IEEE International Conference on Image Processing (ICIP 95)*, Arlington, VA, USA, Oct 1995.

[20] S. Jacobs and A. Eleftheriadis, "Streaming Video Using Dynamic Rate Shaping and TCP Flow Control," *Visual Communication and Image Representation Journal*, Jan 1998.

[21] E. Viscito and C. Gonzales, "A Video Compression Algorithm with Adaptive Bit Allocation and Quantization," *SPIE Vol. 1605 Visual Communication and Image Processing 91*, 1991.

[22] W. Ding and B. Liu, "Rate Control of MPEG Video Coding and Recording by Rate-Quantization Modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 1, pp. 12-19, Feb 1996.

[23] P. J. Brightwell, S. J. Dancer and M. J Knee, "Flexible Switching and Editing of MPEG-2 Video Bitsreams," *International Broadcasting Convention*, Amsterdam, Netherlands, pp. 547-552, Sep 1997.

[24] J. Meng and S.-F. Chang, "CVEPS – A Compressed Video Editing and Parsing System," *Proceedings of ACM Multimedia 96*, Boston, MA, USA, Nov 1996.

[25] B. G. Haskell, A. Puri and A. N. Netravali, "Digital Video: An Introduction to MPEG-2," *Chapman & Hall*, 1997.

[26] C.-Y. Lin and S.-F. Chang, "Issues and Solutions for Video Content Authentication," *CU/ADVENT Technical Report*, Jan 1999.