On Security of Content-Based Video Stream Authentication

Swee-Won Lo(✉), Zhuo Wei, Robert H. Deng, and Xuhua Ding

School of Information Systems,
Singapore Management University, Singapore, Singapore
{sweewon.lo.2009,robertdeng,xhding}@smu.edu.sg

Abstract. Content-based authentication (CBA) schemes are used to authenticate multimedia streams while allowing content-preserving manipulations such as bit-rate transcoding. In this paper, we survey and classify existing transform-domain CBA schemes for videos into two categories, and point out that in contrary to CBA for images, there exists a common design flaw in these schemes. We present the principles (based on video coding concept) on how the flaw can be exploited to mount semantic-changing attacks in the transform domain that cannot be detected by existing CBA schemes. We show attack examples including content removal, modification and insertion attacks. Noting that these CBA schemes are designed at the macroblock level, we discuss, from the attacker's point of view, the conditions in attacking content-based authenticated macroblocks.

Keywords: Content-based authentication · Attack · H.264/AVC

1 Introduction

Video editing tools widely used for synthesizing videos are often being used to maliciously manipulate content of videos for commercial or political purposes, or with the intention to evade law (e.g. surveillance streams). Without an authentication mechanism in place, both the sending and receiving entities could not verify the integrity of a video transmitted through open and insecure networks.

There are two general approaches for multimedia authentication [36], namely Cryptographic-Based Authentication (CrBA) and Content-Based Authentication (CBA). As its name suggests, CrBA schemes (e.g. [8,43]) use cryptographic techniques such as hash function and digital signature algorithm to compute authentication data for the multimedia object. To verify its integrity, a verifier recomputes the hash of the object and verifies it against the digital signature. One of the shortcomings of CrBA schemes is that they are sensitive to random errors due to lossy networks; they are also unable to authenticate objects that usually undergo content-preserving manipulations such as bit-rate transcoding. A CBA scheme (e.g., [5,32,41]) authenticates the semantic meaning of a multimedia object by extracting an invariant *feature* from the object and computing

G. Pernul et al. (Eds.): ESORICS 2015, Part I, LNCS 9326, pp. 366–383, 2015.
DOI: 10.1007/978-3-319-24174-6_19

authentication data (using keyed-hash function or digital signature algorithm) on the feature. The integrity of the object can be verified as long as its feature (i.e., semantic meaning) is unchanged. Hence, CBA schemes are more error-tolerant as compared to CrBA schemes and they allow content-preserving manipulations on the object; some also have the ability to localize tampered regions.

Existing CBA schemes for videos may perform feature extraction in either the pixel, transform or bitstream domain. In this paper, we focus on transform-domain CBA schemes for the following reasons: Although pixel-domain feature extraction is robust to content-preserving manipulations and random errors, it is more computational intensive since the verifier needs to fully decode the video before verification. In retrospect, bitstream-domain feature extraction is more efficient, but it is sensitive to random errors and content-preserving manipulations; since the authentication data is embedded at the bitstream level, it also affects the video quality. Transform-domain CBA schemes are designed to trade off between efficiency, error-tolerance and video quality. In a nutshell, in a transform-domain CBA scheme, the authenticator extracts an invariant transform-domain feature and computes the authentication data, which is embedded back to the video as a watermark. During verification, a verifier extracts feature from the received video, and verifies it against the extracted authentication data (i.e., watermark). Thus, both feature extraction and watermark extraction are pivotal in a CBA scheme to ensure that a video can be securely authenticated.

It is worth noting that earlier work on CBA first focused on the authentication of still images. For example in [2] and [11], the CBA scheme extracts and authenticates feature from the transform coefficients of JPEG and JPEG 2000 images, respectively. These schemes have been proven to be highly efficient and are able to detect semantic-changing attacks. Since video is a sequence of frames, and each frame is essentially a still image, many existing CBA schemes for video adopt a similar design convention as that for images. The work of [5,14,32], for example, extract a transform-domain feature from the frame's coefficients (hereinafter called the *payload*), and show that the feature can detect semantic-changing attacks while remain unchanged under bit-rate transcoding. For applications such as surveillance videos that may lose vital details if transcoded, the work of [10,25] extract a fragile feature from the frame *header* and show that both semantic-changing attacks and bit-rate transcoding cause an avalanche change on the header parameters that inevitably destroys the feature.

The work of [7] summarizes three most common security problems in CBA schemes for images, namely undetected modifications, information leakage and protocol weakness. For example, in [9], the authors point out that due to independent pixel-/block-wise feature and watermark extraction, the schemes in [37] and [40] are vulnerable to *collage attacks*; an attacker can swap pixels or blocks within an image (or among database of images authenticated using the same secret key) to produce a counterfeit image. To thwart this attack, [18] proposes to extract feature from one block and embed its watermark into another randomly selected block. However, due to information leakage in watermark generation, the secret block

relationship graph can be exposed by an attacker [3]. A similar flaw in [16,17] has also been exploited by [38] to expose the secret relationship graph via a *verification device attack* [7]. While there are many studies on security of CBA schemes for images, not many have been done for videos. To the best of our knowledge, the work that addresses security of CBA schemes for video is that of [34], where a flaw in watermark generation in [28] is identified.

The main focus of this paper is to study the security of existing transform-domain CBA schemes as a mean to integrity-protect videos transmitted through open and insecure network. We first survey and categorize existing transform-domain CBA schemes into two categories, namely header- and payload-protected CBA schemes, and we point out a common design flaw in these schemes: the transform-domain feature extracted and authenticated in these schemes is insufficient to securely authenticate a video. We note that while both categories of schemes are able to detect semantic-changing attacks performed in the pixel domain, they are unable to detect attacks performed in the transform domain. We show that unlike images, where the payload (coefficients) represent its overall semantic meaning, the payload and the header of a video have a strong inter-dependency relationship. This relationship, when maliciously exploited, changes the semantic meaning of the final, decoded video to a similar effect as attacks in the pixel domain, and these attacks cannot be detected by the CBA schemes. We discuss the ways that the relationship can be exploited and we show several attack examples (some of which were given in [19]). Finally, we discuss in depth the attacks that manipulate the header of a video, and the conditions of the attack, given the attacker's desired attack content. Note that although our attacks are performed on H.264/AVC-encoded videos, they are also applicable to CBA-protected videos encoded by other standards such as MPEG-2, MPEG-4 and H.264/SVC due to the same underlying video coding concept.

2 The H.264 Video Coding Standard

Most video coding standards including MPEG-2, MPEG-4 and H.264 achieve compression by identifying similarities in the spatial (within frame) and the temporal (between frames) dimensions. In the H.264 standard, a prediction model takes as input a raw video frame and outputs a residual frame. The raw frame is first partitioned into units (each of size 16×16 pixels) called macroblocks, which may be further partitioned into 16 (4×4)-blocks. Given a raw macroblock Ori, the prediction model searches for the most perceptually similar macroblock within a *searchable region*, i.e., neighbouring macroblocks in the same frame (intra prediction) or in adjacent frames (inter-prediction), and uses the most similar macroblock as reference to generate a prediction macroblock Pred. The prediction macroblock is (pixel-wise) subtracted from the raw macroblock to obtain the residual macroblock Res as in (1). The residual macroblock is then transformed, quantized and entropy encoded to the bitstream domain.

Figure 1 shows the syntax of a H.264 macroblock. In this figure, parameter *type* indicates whether the macroblock is intra- or inter-predicted.

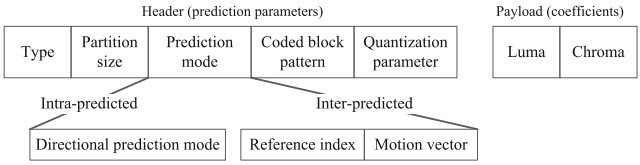


Fig. 1. The syntax elements of a H.264 macroblock.

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

At the decoder, the decoded macroblock Dec is obtained as in (2) after reconstructing the prediction macroblock Pred^* (using the macroblock header) and the residual macroblock Res^* (using the macroblock payload). Note that for a non-tampered macroblock, the quantity α is due to lossy compression and is negligible, and Dec is perceptually similar to Ori. We can also observe an inter-dependent relationship between the macroblock header and payload from (2).

$$\text{Encoder: } \mathsf{Res} = \mathsf{Ori} - \mathsf{Pred} \tag{1}$$

$$\text{Decoder: } \mathsf{Dec} = \mathsf{Pred}^* + \mathsf{Res}^* = \mathsf{Ori} + \alpha \tag{2}$$

In the next section, we show the attacks that can be performed on each category by exploiting the relationship between macroblock header and its payload.

3 Common Design Flaw in Existing Content-Based Video Authentication Schemes

We describe a generic CBA model which is followed by most of the CBA schemes in the literature and classify existing schemes based on the domain of feature extraction in Sects. 3.1 and 3.2, respectively. The design flaw of the CBA schemes is described in detail in Sect. 3.3 and we show how the flaw can be exploited to achieve semantic-changing attacks without being detected by the CBA schemes.

3.1 Content-Based Authentication Model

A transform-domain CBA scheme for video works at the *macroblock* level, in compatible with video coding standards such as MPEG-2, MPEG-4 and H.264

that use block-based coding. Given a macroblock in the transform domain, a CBA scheme first identifies the feature extraction domain and the prediction parameter(s) or coefficients to extract feature F from. The feature F, together with the authenticator's private key sk, serve as inputs to the feature authentication phase that outputs a watermark W_F. In the watermark embedding phase, a different secret key k is used to identify a set of embedding locations and W_F is embedded into the macroblock following a set of embedding rules. The watermarked macroblock is then entropy encoded into a bitstream and transmitted to a verifier. Upon receiving the bitstream, the verifier performs entropy-decoding and watermark extraction by identifying the extraction domain, locations and extraction rules to output the watermark W_F. The verifier then performs the same feature extraction operation to output a feature F' of the macroblock and verifies it against W_F using the authenticator's public key pk (which corresponds to the authenticator's private key sk) in the feature verification phase. Upon successful verification, the verifier proceeds to decode the macroblock.

3.2 Classification of Existing Schemes

We classify existing CBA schemes for video into two categories, namely payload- and header-protected CBA schemes.

Payload-Protected Schemes. Payload-protected schemes extract and authenticate a feature from the macroblock payload (i.e., coefficients) that is able to detect semantic-changing attacks and survive bit-rate transcoding. The watermark computed from the feature is embedded back into the coefficients in the payload, or into the prediction parameters in the header. For embedding into payload, the rule of evaluating LSB [5,32,35,41], zero/non-zero coefficients [42] or energy relationship between coefficients [4,35] are used, whereas for embedding into header, the rule of evaluating LSB [14,28] of MVs is used.

Header-Protected Schemes. In the work of [10,25], a feature is extracted, respectively, from the DPMs of intra frames and the partition sizes of macroblocks. Their schemes are shown to reliably detect semantic-changing attacks as well as unauthorized bit-rate transcoding due to the fragile nature of header parameters. The watermark is embedded into the payload using the LSB evaluation rule due to limited embedding capacity in the header.

Remark. We note that there are several CBA schemes that extract feature from the payload *and* motion vectors [15,30,41] in the header. For clarity sake, we do not classify them but as we will discuss and show in the remainder of the paper, almost all prediction parameters in the header have interdependent relationship with the payload that can be exploited to achieve semantic-changing attacks; these schemes are still susceptible to our attacks in the transform domain.

3.3 The Design Flaw and Its Exploitation

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

Exploiting the Flaw in Payload-Protected Schemes. Unlike images where image pixels were directly transformed and quantized [31], a video's macroblock coefficients convey the *relationship* between the macroblock pixel content and its prediction macroblock, i.e., the residual macroblock Res. If an attacker finds an *attack prediction macroblock* Pred' to replace the *original prediction macroblock* Pred^*, the targeted macroblock Dec could be modified to the attacker's desired attack macroblock Dec' (see (2)). Hereafter, we base our discussion at the (4×4)-block level since it is the smallest coding unit.

To find an attack prediction block, an attacker proceeds as follow. Firstly, identify the "searchable region" and the candidate reference blocks that generate the suitable Pred' to obtain Dec'. In intra-prediction, the searchable region is the four neighbouring blocks (left, above-and-to-the-left, above, and above-and-to-the-right of) the targeted block whereas in inter-prediction, the searchable region is within an area centering on the targeted block [44]. To replace Pred^* with Pred', modify the *prediction mode* (e.g., DPM, MV and reference frame index) of the targeted block Dec.

Depending on the video content, it is possible that a suitable Pred' is unavailable. If so, a workaround that indirectly modifies the residual block Res without being detected by payload-protected schemes can be performed using the effect of QP. At the encoder, a larger QP in forward quantization removes insignificant coefficients. At the decoder, given a set of coefficients, a larger QP in inverse quantization magnifies the residual samples whereas a smaller QP suppresses the samples. If a decoder receives a corrupted QP, inverse quantization results in a different set of coefficients that may misrepresent the residual samples in Res. Note that this cannot be detected by payload-protected schemes because the magnifying/suppressing happens during the decoding process, which is only executed *after* integrity verification. Having different QPs across macroblocks in a frame is not uncommon; macroblock-layer rate control in H.264 has been proven to improve coding efficiency [21] whereas earlier standards (e.g., MPEG-4) and the H.264 High Profile allow different QPs for DC and AC coefficients [6,27].

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

sub-blocks, such that the targeted content is isolated in a sub-block, and then perform a search for the suitable attack prediction sub-block thereof.

Remarks. Attacks on payload-protected schemes involve replacing the original prediction block with an attack prediction block in order to change the content of a targeted block. Given the searchable region which is constrained in one frame (intra frames) or within the same video (inter frames), arbitrary content insertion attacks cannot be realized. However, content removal and modification attacks are possible as will be shown in Sect. 4. We also note that prediction mode parameters such as DPM, MV and reference frame index are coded differentially between successive blocks. If these parameters are changed, it may affect the corresponding parameter of subsequent (targeted/non-targeted) blocks, causing them to use a wrong/different prediction block for decoding. This may result in error propagation that occur in the form of visual distortion on the decoded frame. In Sect. 4, we show an example of such error propagation, and show that the visual distortion can be corrected to a certain degree by either restoring the prediction mode of affected blocks, or by restricting their choice of prediction block to a more suitable one.

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

Complying with Watermark Extraction. Apart from ensuring that the transform-domain attacks do not alter the authenticated feature, it is also vital to ensure that the tampered data obeys the watermark extraction rule. Watermark extraction includes: extract location identification and extraction based on extraction rules. Although random extraction locations is deemed vital for security reason [22], we argue that it is more important for copyright protection where the attack objective is to find and destroy the watermark; a successful attack in our approach depends more heavily on complying with the watermark extraction rules. For verification efficiency, existing CBA schemes perform extraction by evaluating either the LSB or zero/non-zero coefficients as mentioned in Sect. 3.2. Such characteristics can always be engineered in the coefficients or MVs. Since DPMs can be categorized into sets generating similar prediction blocks [24], an attacker can select DPMs within the same set to satisfy the even/odd evaluation.

4 Attack Examples on Existing CBA Schemes

In this section, we demonstrate transform-domain attack examples[1] that can be applied on each category of CBA schemes as discussed in Sect. 3.3. More specifically, we show content removal and content modification attacks on payload-protected schemes, and content insertion attack on header-protected schemes. Our attacks are implemented using the JM reference software [12]. We emulate the attacker's interception and replacement of macroblock stream by modifying the decoder's 'read' data. The video sequences used in our attacks are the 352×288 News, Bridge and Waterfall sequences [1] and a 384×288 surveillance sequence [33], all encoded in IBBBBBBBP format with $QP = 28$ for intra frames and $QP = 30$ for inter frames.

4.1 Content Removal Attacks

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

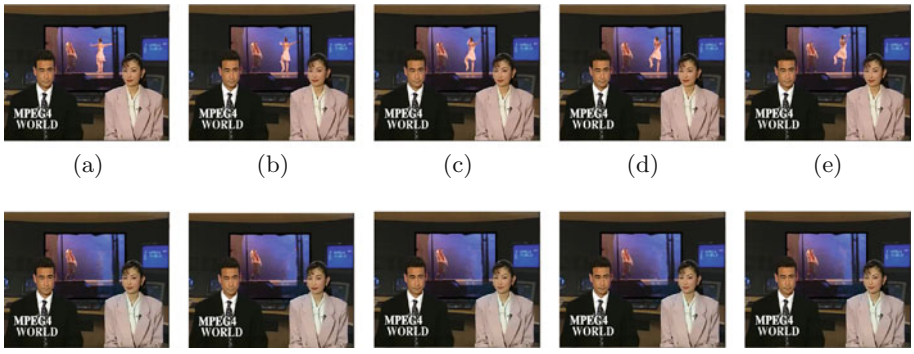
Figure 2a, b, c, d, and e show the first five frames of the original News sequence, where Fig. 2a is an intra frame and Fig. 2b, c, d, and e are inter frames. The aim of the attack is to remove content of the targeted blocks, i.e., the ballerina, by finding new attack prediction blocks such that the end result is a set of attacked blocks that convey the background information, i.e., the walls. Notice that in Fig. 2a, the targeted blocks are surrounded by reference blocks conveying similar content, i.e., the walls. This is an example where the attack prediction blocks *are* the original prediction blocks and it implies that the samples in the (targeted) residual blocks have high magnitude since they do not have similar prediction blocks to be used for compression (see (1)). Hence, the workaround by manipulating QP of the targeted blocks to suppress their residual samples is executed. Subsequently, if necessary, the DPMs of the targeted blocks (e.g., torso of the ballerina) are modified to use background blocks as attack prediction blocks.

Since intra frames are used as (one of the) reference frame(s) to generate prediction blocks for the subsequent inter frames, the content of the attacked intra frame will "propagate" to the inter frames during decoding. The residuals of the original content in inter frames were completely removed by modifying the MV of targeted blocks in the inter frames. The final result of the removal attack is shown in Fig. 2f, g, h, i, and j.

Due to differential coding of prediction mode parameters, there is a risk of error propagation after one is manipulated. Figure 3a shows an example of error propagation due to erroneous DPM decoding in an intra frame, which is resolved by correcting the DPM of the affected block(s) to use a *more* suitable prediction block for decoding. The result of this correction is shown in Fig. 3b.

There are also cases where QP manipulation is not needed. Figure 4a shows the first frame of the Bridge sequence. In this example, it is sufficient to modify

[1] Source files can be viewed at https://sites.google.com/site/smusvc/Authentication.



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

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

Fig. 2. Content removal attack on News sequence, with the original frames shown in (a)–(e) and attack frames in (f)–(j).

(a) Distorted frame (b) Corrected frame

Fig. 3. An example of visual distortion due to DPM decoding error and its corrected version.

(a) Original frame (b) Attacked frame

Fig. 4. Content removal attack on Bridge sequence.

the DPMs of targeted blocks, i.e., the left pier, to use the background information, i.e., the river, as attack prediction blocks. The result of the removal attack is shown in Fig. 4b. In this case, QP manipulation is not needed because the original prediction blocks are obtained from the top of the targeted blocks and they are semantically similar, thus, the residual blocks have samples of small magnitude. Replacing the original prediction blocks with the attack prediction blocks on the left (i.e., the river) replaces the content of the targeted blocks with the content of the attack prediction blocks.

(a) Original frame (b) Attacked frame

Fig. 5. Content replacement attack on Waterfall sequence.

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

Fig. 6. Content replacement attack on a surveillance sequence.

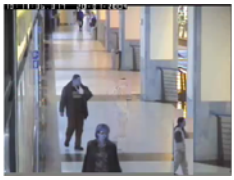
4.2 Content Modification Attacks

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

Content replacement is the act of replacing (or "overwriting") the content of a targeted block with that of its desired attack block. In our first example, we perform content replacement attack on the intra frame of the Waterfall sequence. As shown in Fig. 5, the DPMs of a large set of targeted blocks are modified to "extend" the effect of attack prediction blocks, i.e., the trees, such that they cover the original blocks, i.e., the waterfall.

In the second example, the reference frame index is modified to achieve content replacement in inter frames. In addition, the partition size parameter is also modified to facilitate the attack. Figure 6b shows the timing information extracted from a surveillance frame in Fig. 6a. This timing information is encoded using 16×16 macroblocks, where the upper half of each macroblock covers the timing information (targeted) while the lower half covers the surveillance background (non-targeted). Tampering with the reference frame index will affect *both* the timing information and the surveillance background. By manipulating the *partition size* parameter such that each targeted 16×16 macroblock is partitioned into sixteen 4×4 blocks, the targeted content is isolated from the non-targeted content. The reference frame index of the targeted blocks can then be modified independently without affecting the non-targeted blocks. Figure 6c shows the result of this attack; when the attacked frames are inserted into the video sequence, a scrambled timing information is observed.

(a) Original frame



(b) Tampered frame

Fig. 7. Content relocation attack on a surveillance sequence.

Fig. 8. Content insertion attack on header-protected CBA schemes.

Content relocation is the act of changing the position of an object from one location to another. This attack is typically difficult to achieve on intra frames because each intra block is predicted from its neighbouring blocks; to perform a meaningful content relocation that is affected by, and will be affecting, neighbouring blocks is intuitively hard. For an inter block, however, this attack can be achieved by modifying the MV using a concept similar to content removal. Figure 7a shows a frame extracted from the surveillance sequence. In this attack, the MVs of the targeted blocks, i.e., the dustbin, are modified such that they use a new content, i.e., the man, as attack prediction blocks. Subsequently, the blocks containing the man is removed using the content removal attack methodology presented in the previous subsection. The result of this attack is shown in Fig. 7b.

4.3 Content Insertion Attacks

For completeness, we show an example of content insertion attack on header-protected schemes since this attack is not possible on payload-protected schemes. Figure 8 shows an example of content insertion attack on header-protected CBA schemes, where the original frame is shown in Fig. 6a. Taking the samples of arbitrary image of a clock, the residual blocks are obtained by subtracting the original prediction blocks from the samples. The residual blocks are then transformed and quantized, and inserted into the macroblock payload.

4.4 Summary and Remarks

In summary, we showed that contrary to images, the video header and payload must be simultaneously integrity protected since their interdependency relationship can be exploited by attacks performed in the transform domain. For attacks

on payload-protected schemes, DPM, MV and reference frame index affect the generation of prediction block, which when combined with the residual block could semantically change the targeted block. While DPM selects prediction blocks from neighbouring blocks, MV and reference frame index select them from a wider search range. An advanced attacker may modify the macroblock type (intra/inter) and remove or insert bogus prediction mode relevant to the new macroblock type; we leave attacks of such nature as future work. Additionally, the QP is a header parameter that can be used as a workaround to inexplicitly modify the residual block while the partition size can be modified to facilitate search for a suitable prediction block. For attacks on header-protected schemes, it is vital that the distribution of tampered coefficients tallies with the coded block pattern (CBP) in the header, otherwise a decoding error may occur. We acknowledge that authenticating the CBP in the header will impose a higher level of difficulty on the attacks, however, in existing header-protected schemes, this parameter is often left unprotected. In the literature, there are also CBA schemes that authenticate both the payload and the MVs in the header [15,30,41]. However, as we have shown in our attack examples, these schemes are still vulnerable to attacks such as DPM attacks on intra blocks, reference frame index and/or partition size attacks on inter blocks.

We also note some interesting observations on H.264/SVC - the scalable extension of H.264/AVC that is used to encode the sequences used in this study. In SVC, a mandatory base layer (BL) that is backward compatible with AVC is encoded. Using BL as reference to generate prediction, one or more enhancement layers (ELs) that gradually improve the resolution or quality of the video are encoded. If header-protected CBA schemes are applied on an SVC stream, attacks on the payload of BL and ELs are possible (and powerful). On the other hand, if coefficients-protected CBA schemes are applied, our attacks are applicable to the BL and the effect could propagate to the ELs. Thus, noting the importance of the BL, the work of [36] cryptographically protects the BL to prevent any form of malicious tampering[2]. Although there are minimal header parameters in the ELs [29], we observe the following important parameters, e.g., the *motion prediction flag* and *residual prediction flag*. For the ELs, a motion prediction flag of '1' indicates that the EL directly uses header parameters of its reference (base) layer; otherwise, it carries its own header parameters. A residual prediction flag of '1', on the other hand, indicates that the EL's payload $\mathsf{R}'_{\mathsf{EL}}$ is obtained by subtracting the upsampled BL payload R_{BL} from the payload obtained via AVC-like encoding R_{EL}; otherwise, $\mathsf{R}'_{\mathsf{EL}} = \mathsf{R}_{\mathsf{EL}}$. An advanced attacker could then opt to modify these flags and to manipulate the video semantic. In short, content-based authentication for SVC present several interesting research problems to be explored.

[2] Pixel-domain CBA scheme is used in [36] to protect the ELs and thus is out of the scope of study for this paper.

5 Discussions

We have shown that semantic-changing attacks on videos authenticated by payload- or header-protected CBA schemes are possible by modifying, respectively, the header or the payload of the targeted block(s). Moreover, these attacks cannot be detected by the respective CBA schemes.

Since the attacks on header-protected schemes are relatively straightforward, we focus the following discussions on the attacks on *payload-protected schemes*. As shown in Sect. 4, a targeted block will convey a semantically different content as compared to its original content if a *suitable* attack prediction block is found from the searchable region. In this section, we analyze (from an attacker's point of view) that given the desired attack block and the unmodifiable residual block, whether it is possible for an attacker to obtain the suitable attack prediction block. Our analysis is performed on a 4×4-block level, where a macroblock M is represented as follow:

$$\mathsf{M} = \begin{array}{|c|c|c|c|} \hline \mathsf{M}(\mathsf{B}_0) & \mathsf{M}(\mathsf{B}_1) & \mathsf{M}(\mathsf{B}_4) & \mathsf{M}(\mathsf{B}_5) \\ \hline \mathsf{M}(\mathsf{B}_2) & \mathsf{M}(\mathsf{B}_3) & \mathsf{M}(\mathsf{B}_6) & \mathsf{M}(\mathsf{B}_7) \\ \hline \mathsf{M}(\mathsf{B}_8) & \mathsf{M}(\mathsf{B}_9) & \mathsf{M}(\mathsf{B}_{12}) & \mathsf{M}(\mathsf{B}_{13}) \\ \hline \mathsf{M}(\mathsf{B}_{10}) & \mathsf{M}(\mathsf{B}_{11}) & \mathsf{M}(\mathsf{B}_{14}) & \mathsf{M}(\mathsf{B}_{15}) \\ \hline \end{array}$$

where $\mathsf{M}(\mathsf{B}_i)$ denotes the i-th (4×4)-block of M, and can be represented by a 4×4 matrix. Using the same convention, an original and prediction macroblock (denoted Dec and Pred respectively), are made up of $\mathsf{Dec}(\mathsf{B}_i)$ and $\mathsf{Pred}(\mathsf{B}_i)$ for $i = 0, \cdots, 15$. The list of notations is shown in Table 1.

Generally, the average value of a 4×4-block is a good approximation of the block's samples [20,23,39] due to the high correlation between samples in the block. Since the residual block may consist of positive and negative integers, we

Table 1. List of notations

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

use the median of the residual samples to indicate the relationship between the original block and the original prediction block. In other words, if $\hat{r}(\mathsf{B}) > 0$, then $\mathsf{Dec}(\mathsf{B})$ is perceptually brighter than $\mathsf{Pred}(\mathsf{B})$; otherwise, $\mathsf{Dec}(\mathsf{B})$ is perceptually darker than $\mathsf{Pred}(\mathsf{B})$. In addition, we let $\mathsf{E}(\mathsf{Res})$ be the energy of the residual samples in $\mathsf{Res}(\mathsf{B})$ as defined in Table 1.

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

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

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

$$\begin{aligned} \bar{p}(\mathsf{B}) + \hat{r}(\mathsf{B}) &< \bar{p}'(\mathsf{B}) + \hat{r}(\mathsf{B}) < \bar{p}(\mathsf{B}) + \hat{r}(\mathsf{B}) + 2\hat{r}(\mathsf{B}) \\ \bar{p}(\mathsf{B}) &< \quad \bar{p}'(\mathsf{B}) \quad < \bar{p}(\mathsf{B}) + 2\hat{r}(\mathsf{B}) \end{aligned} \tag{3}$$

Referring to (3), we say that an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ with $\bar{p}'(\mathsf{B})$ cannot be found. Otherwise, by computing $\mathsf{Res}'(\mathsf{B}) = \mathsf{Dec}(\mathsf{B}) \ominus \mathsf{Pred}'(\mathsf{B})$, the upper and lower bound of $\hat{r}'(\mathsf{B})$ is:

$$\begin{aligned} \hat{r}'(\mathsf{B})_{UB} &= \bar{d}(\mathsf{B}) - \bar{p}(\mathsf{B}) = \hat{r}(\mathsf{B}), \text{ and} \\ \hat{r}'(\mathsf{B})_{LB} &= \bar{d}(\mathsf{B}) - (\bar{p}(\mathsf{B}) + 2\hat{r}(\mathsf{B})) = -\hat{r}(\mathsf{B}) \end{aligned}$$

In other words, $-\hat{r}(\mathsf{B}) < \hat{r}'(\mathsf{B}) < \hat{r}(\mathsf{B})$. This implies that compared to the original prediction block $\mathsf{Pred}(\mathsf{B})$, the attack prediction block $\mathsf{Pred}'(\mathsf{B})$ generates smaller residual samples if it is used to encode $\mathsf{Dec}(\mathsf{B})$. This contradicts the video coding rule, where $\mathsf{Pred}(\mathsf{B})$ is initially chosen to encode $\mathsf{Dec}(\mathsf{B})$ because it generates the smallest Sum of Absolute Errors, $\mathrm{SAE} = \sum_{i,j=0}^{3} |d_{i,j} - p_{i,j}|$, where $d_{i,j}$ is the sample of $\mathsf{Dec}(\mathsf{B})$ at position (i, j) and $p_{i,j}$ is the sample of $\mathsf{Pred}(\mathsf{B})$ at position (i, j), compared to all other candidate prediction blocks in the searchable region [27]. This case can be demonstrated in the attack shown in Fig. 4b. If $\mathsf{Pred}'(\mathsf{B})$ cannot be found, the workaround by manipulating QP can be implemented to suppress the residual samples so that the available candidate prediction blocks can be used to obtain $\mathsf{Dec}'(\mathsf{B})$.

Case 1B. When most of the residual samples are positive, i.e., $\hat{r}(\mathsf{B}) > 0$, but the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually darker than the original block $\mathsf{Dec}(\mathsf{B})$, then the minimum value for a sample $d'_{i,j}$ in $\mathsf{Dec}'(\mathsf{B})$ must be equal to the residual sample $r_{i,j}$ in $\mathsf{Res}(\mathsf{B})$. This is because the minimum sample for $\mathsf{Dec}'(\mathsf{B})$ is when $\mathsf{Pred}'(\mathsf{B}) = 0$ (see (2)), otherwise, if $\bar{d}'(\mathsf{B}) < \hat{r}(\mathsf{B})$, then by substituting (2),

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

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

we get $\bar{p}'(\mathsf{B}) + \hat{r}(\mathsf{B}) < \hat{r}(\mathsf{B})$ and the samples in the attack prediction block is less than zero, which is not feasible. Thus, we write, in approximation, $\hat{r}(\mathsf{B}) \leq \bar{d}'(\mathsf{B}) < \bar{d}(\mathsf{B})$. This condition is demonstrated in the attack shown in Fig. 2, where the background information (the walls) is perceptually darker than the targeted blocks (the ballerina), but the residuals samples are too large for $\mathsf{Dec}'(\mathsf{B})$ to satisfy this condition. The QP can then be manipulated to suppress/magnify the residual samples as deemed necessary.

Case 2A. When most of the residual samples are negative, i.e., $\hat{r}(\mathsf{B}) < 0$, the original block $\mathsf{Dec}(\mathsf{B})$ is perceptually darker than the original prediction block $\mathsf{Pred}(\mathsf{B})$. Suppose the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually brighter than $\mathsf{Dec}(\mathsf{B})$, we say that a sample $d'_{i,j}$ in $\mathsf{Dec}'(\mathsf{B})$ is upper bounded by $255 - |r_{i,j}|$ in $\mathsf{Res}(\mathsf{B})$ as dictated by (2). Thus, we can write in approximation that $\bar{d}(\mathsf{B}) < \bar{d}'(\mathsf{B}) \leq 255 - |\hat{r}(\mathsf{B})|$. A similar analysis to Case 1B can be applied, where if $\bar{d}'(\mathsf{B}) > 255 - |\hat{r}(\mathsf{B})|$, then the attacker must find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ where $\bar{p}'(\mathsf{B}) > 255$, which is not possible. This condition can be observed in the attack shown in Fig. 5b.

Case 2B. When most of the residual samples are negative, i.e., $\hat{r}(\mathsf{B}) < 0$, but the desired attack block $\mathsf{Dec}'(\mathsf{B})$ is to be perceptually darker than the original block $\mathsf{Dec}(\mathsf{B})$, then we say that an attacker can find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ if and only if $\bar{d}'(\mathsf{B}) \leq \bar{d}(\mathsf{B}) - 2|\hat{r}(\mathsf{B})|$. This condition can be obtained by a similar prove by contradiction as in Case 1A, whereas an illustration example is shown in the upper torso, especially the head of the ballerina in Fig. 2a.

Table 2 summarizes the conditions for the above cases. When the attack block $\mathsf{Dec}'(\mathsf{B})$ cannot satisfy the conditions in any of the cases above, then the attacker cannot find an attack prediction block $\mathsf{Pred}'(\mathsf{B})$ such that $\mathsf{Dec}'(\mathsf{B}) = \mathsf{Pred}'(\mathsf{B}) \oplus \mathsf{Res}(\mathsf{B})$. A workaround can then be performed by modifying the unprotected QP to suppress or magnify the residual samples depending on the available candidate attack prediction blocks.

6 Conclusions and Future Work

We have shown that existing content-based authentication (CBA) schemes designed for videos are insecure due to insufficient feature extraction. The overlooked interdependent relationship between the header and payload parameters can be exploited to perform semantic-changing attacks in the transform domain. We showed several semantic-changing attack examples that are performed in the

transform domain and our attacks cannot be detected by the schemes. We discussed in detail the conditions at which an attack on payload-protected CBA schemes can succeed given a desired attack content and the unmodifiable payload, and if not, a workaround for it.

A possible countermeasure to our attacks is to use more complicated watermark extraction rules. However, unlike images, real-time extraction is vital for video authentication [26] which makes straight-forward watermark extraction rules such as those surveyed in this paper highly preferred. Another possible countermeasure is to extract and authenticate features from both the header and payload domains. In practical applications, transcoding requires full decoding of intra frames and partial decoding of inter frames. The transcoding of intra frames drastically changes the header and payload [13], and to the best of our knowledge, there is no work that addresses this problem. Transcoding inter frames changes the payload while the remaining data in the header remains unchanged. Although existing payload-domain schemes are able to extract a stable feature from the coefficients, but sparsely-distributed coefficients in inter frames (especially after transcoding) are commonly overlooked, thereby leaving them vulnerable to tampering. Thus far, we observed a stable feature from the header of intra frames and we are working on extracting a stable feature from the header of inter frames. Our future research is to design a secure and efficient authentication scheme that overcomes the vulnerability of existing schemes and is robust against bit-rate transcoding (performed by semi-trusted intermediary proxies) as described above.

Acknowledgement. This work was partly supported by National Natural Science Funds of China (Grant No. 61402199).

References

1. Arizona State University: Video trace library. http://trace.eas.asu.edu/index.html
2. Bianchi, T., Rosa, A.D., Piva, A.: Improved DCT coefficient analysis for forgery localization in JPEG images. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011), pp. 2444–2447 (2011)
3. Chang, C., Fan, Y., Tai, W.: Four-scanning attack on hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recogn. **41**(2), 654–661 (2008)
4. Dai, Y., Thiemert, S., Steinebach, M.: Feature-based watermarking scheme for MPEG-I/II video authentication. In: SPIE Security, Steganography, and Watermarking of Multimedia Contents VI, vol. 5306, pp. 325–335 (2004)
5. Du, R., Fridrich, J.: Lossless authentication of MPEG-2 video. In: 2002 International Conference on Image Processing (ICIP), vol. 2, pp. II-893–II-896 (2002)
6. Ebrahimi, T., Horne, C.: MPEG-4 natural video coding - an overview. Signal Process. Image Commun. **15**(4–5), 365–385 (2000)
7. Fridrich, J.: Security of fragile authentication watermarks with localization. In: SPIE, Security and Watermarking of Multimedia Contents IV, vol. 4675, pp. 691–700 (2002)

8. Hefeeda, M., Mokhtarian, K.: Authentication schemes for multimedia streams: quantitative analysis and comparison. ACM Trans. Multimedia Comput. Commun. Appl. **6**(1), 1–24 (2010)
9. Holliman, M., Memon, N.: Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. IEEE Trans. Image Process. **9**(3), 432–442 (2000)
10. Horng, S.J., Farfoura, M.E., Fan, P., Wang, X., Li, T., Guo, J.M.: A low cost fragile watermarking scheme in H.264/AVC compressed domain. Multimedia Tools Appl. **72**(3), 2469–2495 (2014)
11. Hu, H.T., Hsu, L.Y.: Exploring DWT-SVD-DCT feature parameters for robust multiple watermarking against JPEG and JPEG2000 compression. Comput. Electr. Eng. **41**, 52–63 (2015)
12. Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6).: JM Reference Software Manual (2009)
13. Kim, D., Choi, Y., Kim, H., Yoo, J., Choi, H., Seo, Y.: The problems in digital watermarking into intra-frames of H.264/AVC. Image Vis. Comput. **28**(8), 1220–1228 (2010)
14. Kuo, T.Y., Lo, Y.C., Lin, C.I.: Fragile video watermarking technique by motion field embedding with rate-distortion minimization. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP), pp. 853–856 (2008)
15. Lin, C.Y., Chang, S.F.: Issues and solutions for authenticating MPEG video. In: SPIE International Conference on Security and Watermarking of Multimedia Contents, vol. 3657, pp. 54–56 (1999)
16. Lin, C.Y., Chang, S.F.: Semi-fragile watermarking for authenticating JPEG visual content. In: SPIE Security and Watermarking of Multimedia Contents, pp. 140–151 (2000)
17. Lin, C.Y., Chang, S.F.: A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Trans. Circuits Syst. Video Technol. **11**(2), 153–168 (2001)
18. Lin, P., Hsieh, C., Huang, P.: A hierarchical digital watermarking method for image tamper detection and recovery. Pattern Recogn. **38**(12), 2519–2529 (2005)
19. Lo, S.W., Wei, Z., Deng, R.H., Ding, X.: Generic attacks on content-based video stream authentication. In: IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6 (2014)
20. Luo, Z., Song, L., Zheng, S., Ling, N.: H.264 advanced video control perceptual optimization coding based on JND-directed coefficient suppression. IEEE Trans. Circuits Syst. Video Technol. **23**(6), 935–948 (2013)
21. Ma, S., Gao, W., Zhao, D., Lu, Y.: A study on the quantization scheme in H.264/AVC and its application to rate control. In: Advances in Multimedia Information Processing, PCM, pp. 192–199 (2004)
22. Mansouri, A., Aznaveh, A.M., Torkamani-Azar, F., Kurugollu, F.: A low complexity video watermarking in H.264 compressed domain. IEEE Trans. Inf. Forensics Secur. **5**(4), 649–657 (2010)
23. Naccari, M., Pereira, F.: Advanced H.264/AVC-based perceptual video coding: architecture, tools and assessment. IEEE Trans. Circuits Syst. Video Technol. **21**(6), 806–819 (2011)
24. Park, J.S., Song, H.J.: Selective intra prediction mode decision for H.264/AVC encoders. World Acad. Sci. Eng. Technol. **13**, 51–55 (2006)
25. Park, S.W., Shin, S.: Authentication and copyright protection scheme for H.264/AVC and SVC. J. Inf. Sci. Eng. **27**(1), 129–142 (2011)

26. Podilchuk, C.I., Delp, E.J.: Digital watermarking: algorithms and applications. IEEE Signal Process. Mag. **18**(4), 33–46 (2001)
27. Richardson, I.E.: The H.264 Advanced Video Compression Standard, 2nd edn. Wiley, Chichester (2010)
28. Saadi, K.A., Bouridane, A., Guessoum, A.: Combined fragile watermark and digital signature for H.264/AVC video authentication. In: 17th European Signal Processing Conference, pp. 1799–1803 (2009)
29. Schwarz, H., Merpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst. Video Technol. **17**(9), 1103–1120 (2007)
30. Shahabuddin, S., Iqbal, R., Shirmohammadi, S., Zhao, J.: Compressed-domain temporal adaptation-resilient watermarking for H.264 video authentication. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1752–1755 (2009)
31. Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. IEEE Signal Process. Mag. **18**(5), 36–58 (2001)
32. Sun, Q., He, D., Tian, Q.: A secure and robust authentication scheme for video transcoding. IEEE Trans. Circuits Syst. Video Technol. **16**(10), 1232–1244 (2006)
33. The CAVIAR team: EC funded CAVIAR project/IST 2001 37540. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
34. Ting, G.C., Goi, B.M., Lee, S.W.: Cryptanalysis of a fragile watermark based H.264/AVC video authentication scheme. Appl. Mech. Mater. **145**, 552–556 (2011)
35. Wang, Y., Pearmain, A.: Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain. IEEE Trans. Image Process. **15**(6), 1536–1543 (2006)
36. Wei, Z., Wu, Y., Deng, R., Ding, X.: A hybrid scheme for authenticating scalable video codestreams. IEEE Trans. Inf. Forensics Secur. **9**(4), 543–553 (2014)
37. Wong, P.: A watermark for image integrity and ownership verification. In: IS and TS PICS Conference, pp. 374–379 (1998)
38. Wu, Y., Xu, C.: A fault-induced attack to semi-fragile image authentiation schemes. In: SPIE on Visual Communications and Image Processing, vol. 5150, pp. 1875–1883 (2003)
39. Yang, X., Lin, W., Lu, Z., Ong, E., Yao, S.: Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile. IEEE Trans. Circuits Syst. Video Technol. **15**(6), 742–752 (2005)
40. Yeung, M., Mintzer, F.: An invisible watermarking technique for image verification. In: International Conference on Image Processsing (ICIP), pp. 680–683 (1997)
41. Yin, P., Yu, H.H.: A semi-fragile watermarking system for MPEG video authentication. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. IV-3461–IV-3464 (2002)
42. Zhang, W., Zhang, R., Liu, X., Wu, C., Niu, X.: A video watermarking algorithm of H.264/AVC for content authentication. J. Networks **7**(8), 1150–1154 (2012)
43. Zhao, Y., Lo, S.W., Deng, R.H., Ding, X.: An improved authentication scheme for H.264/SVC and its performance evaluation over non-stationary wireless mobile networks. In: 6th International Conference on Network and System Security (NSS), pp. 192–206 (2012)
44. Zhao, Z., Liang, P.: A statistical analysis of H.264/AVC FME mode reduction. IEEE Trans. Circuits Syst. Video Technol. **21**(1), 53–61 (2011)