9-2016

Trustworthy authentication on scalable surveillance video with background model support

Zhuo WEI
Huawei's Shield Lab, Singapore

Zheng YAN
Xidian University

Yongdong WU
Institute of Infocomm Research, Singapore

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

DOI: https://doi.org/10.1145/2978573

Published in ACM Transactions on Multimedia Computing, Communications and Applications, September 2016, Volume 12, Issue 4s, Article number 64.
http://doi.org/10.1145/2978573

Trustworthy Authentication on Scalable Surveillance Video with Background Model Support

ZHUO WEI, Huawei's Shield Lab, Singapore
ZHENG YAN, Xidian University, China & Aalto University, Finland
YONGDONG WU, Institute for Infocomm Research, Astar, Singapore
ROBERT HUIJIE DENG, Singapore Management University, Singapore

H.264/SVC (Scalable Video Coding) codestreams, which consist of a single base layer and multiple enhancement layers, are designed for quality, spatial, and temporal scalabilities. They can be transmitted over networks of different bandwidths and seamlessly accessed by various terminal devices. With a huge amount of video surveillance and various devices becoming an integral part of the security infrastructure, the industry is currently starting to use the SVC standard to process digital video for surveillance applications such that clients with different network bandwidth connections and display capabilities can seamlessly access various SVC surveillance (sub)codestreams. In order to guarantee the trustworthiness and integrity of received SVC codestreams, engineers and researchers have proposed several authentication schemes to protect video data. However, existing algorithms cannot simultaneously satisfy both efficiency and robustness for SVC surveillance codestreams. Hence, in this article, a highly efficient and robust authentication scheme, named TrustSSV (Trust Scalable Surveillance Video), is proposed. Based on quality/spatial scalable characteristics of SVC codestreams, TrustSSV combines cryptographic and content-based authentication techniques to authenticate the base layer and enhancement layers, respectively. Based on temporal scalable characteristics of surveillance codestreams, TrustSSV extracts, updates, and authenticates foreground features for each access unit dynamically with background model support. Using SVC test sequences, our experimental results indicate that the scheme is able to distinguish between content-preserving and content-changing manipulations and to pinpoint tampered locations. Compared with existing schemes, the proposed scheme incurs very small computation and communication costs.

CCS Concepts: • **Security and privacy → Trust frameworks**; **Authentication;**

Additional Key Words and Phrases: H.264/SVC, authentication, integrity, surveillance application, background model

ACM Reference Format:
Zhuo Wei, Zheng Yan, Yongdong Wu, and Robert Huijie Deng. 2016. Trustworthy authentication on scalable surveillance video with background model support. ACM Trans. Multimedia Comput. Commun. Appl. 12, 4s, Article 64 (September 2016), 20 pages.
DOI: http://dx.doi.org/10.1145/2978573

This work is supported by National Natural Science Funds of China (Grant No. 61402199) and Natural Science Funds of Guangdong (Grant No. 2015A030310017).
Authors' addresses: Z. Wei, 20 Science Park Road #03-31/32 Teletech Park Singapore Science Park II, Singapore, 117674; email: phdzwei@gmail.com; R. H. Deng, School of Information System, Singapore Management University, 178902; email: robertdeng@smu.edu.sg; Z. Yan, The State Key Lab of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, 710071; email: zheng.yan@aalto.fi; Y. Wu, Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore, 138632; email: wydong@i2r.a-star.edu.sg.

1. INTRODUCTION

The scalable extension of H.264, referred to as Scalable Video Coding (SVC) [Schwarz et al. 2007], consists of a single base layer that is compatible with the H.264 Advance Video Coding (AVC) standard, and multiple enhancement layers that improve the video in one of three scalability dimensions (quality, resolution, and time). SVC codecs adapt to subpar network connections by dropping these enhancement layers in order to reduce the frame rate, resolution, or bandwidth consumption of a picture, which prevents the picture from breaking up. For instance, a mobile phone would receive only the base layer, while a high-definition video conferencing console would receive both the base layer and enhancement layers. With huge video surveillance and various devices becoming an integral part of the security infrastructure, the industry is currently starting to use the SVC standard to process digital video for surveillance applications such that clients with different network bandwidth connections and display capabilities can seamlessly access various SVC surveillance (sub)codestreams. Figure 1 illustrates an indoor SVC surveillance system that can flexibly distribute (sub)codestreams to TVs, tablets, and smartphones over different network bandwidths.

However, with sophisticated multimedia processing tools, any layer of SVC surveillance codestreams can be modified without leaving any visible traces for human eyes [Wei et al. 2010, 2012, 2014b]. Modified surveillance data have virtually no value as legal proofs since doubts would always exist. Thus, a video authentication scheme is required to thwart any unauthorized manipulations by verifying the integrity and source of the data [Zhu et al. 2004; Hefeeda and Mokhtarian 2011]. An authentication scheme for authenticating surveillance codestreams should meet three basic requirements: security, computational efficiency, and communication efficiency. Furthermore, a scheme for authenticating an SVC surveillance codestream should have the following additional properties. First, it preserves the scalability of the original SVC surveillance codestream. That is, it authenticates the original SVC codestream once at the source, but allows verification of various three-dimensional (sub)codestreams (spatial, quality, and temporal scalabilities) at the recipient. Second, it is able to pinpoint the tampered regions if tampering indeed occurred. Third, it is robust or resilient to content-preserving manipulations (e.g., scale/recompression images) that do not change the semantic meaning of a codestream but are sensitive to content-changing manipulations (e.g., removing/replacing/inserting images) that modify the semantic meaning of the codestream. Although engineers and researchers have proposed several authentication solutions for SVC codestreams, existing solutions that are classified into content-based authentication, cryptographic-based authentication, and watermarking-based authentication cannot simultaneously satisfy the aforementioned requirements as authenticating big SVC codestreams.

Content-based authentication [Han and Chu 2010] is independent of the compression operation. It ensures the authenticity of multimedia features. The advantage of content-based authentication is that it can distinguish content-preserving and malicious manipulations. The works most relevant to our research are the AUSSS scheme in Wei et al. [2013] and the hybrid scheme in Wei et al. [2014a], which protect the integrity of the base layer and enhancement layers by employing cryptographic and content-based authentication, respectively. The hybrid scheme [Wei et al. 2014a] does not consider video content, which authenticates all images of SVC codestreams with the same solution, while AUSSS considers video contents. AUSSS and the scheme presented in this article try to avoid repeatedly transmitting content-based features of static fields in order to improve communication and computation efficiencies. AUSSS makes use of coding information (e.g., macroblock type, Coded Block Pattern (*cbp*)) of the base layer to locate "Active" Blocks (ABlocks). However, ABlocks may be bypassed

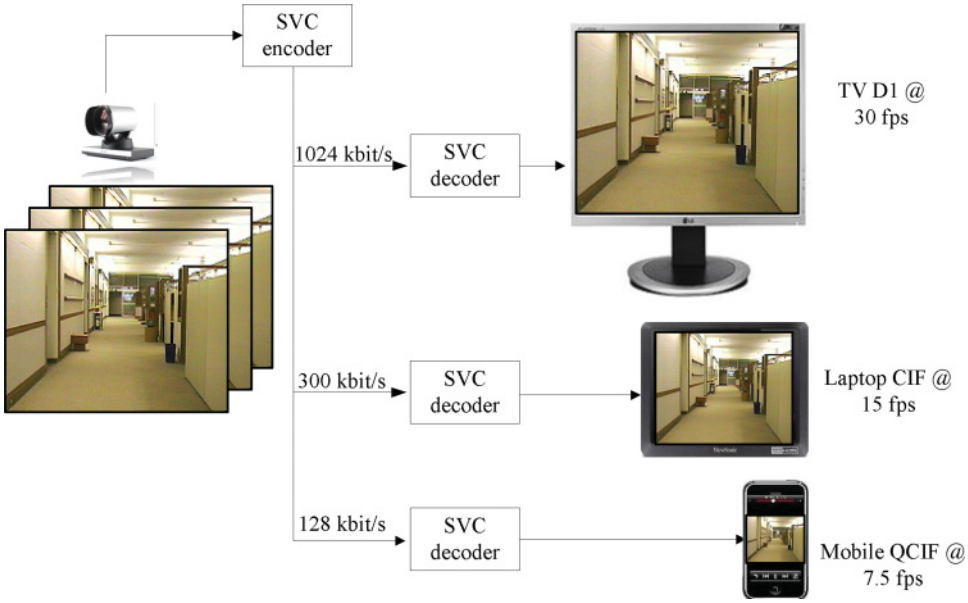


Fig. 1. An example SVC surveillance system.

as the base layer chooses a larger Quantization Parameter (QP).[1] If the QP of the base layer is set too large, coefficients of most of the macroblocks of the base layer are zero, and AUSSS sets the macroblocks as static block. However, the macroblocks in fact contain detailed texture content (i.e., macroblocks should be ABlocks); without extracting and protecting content features of them, adversaries may attacks those macroblocks of the enhancement layer (e.g., removing or replacing content).

Cryptographic-based authentication in general performs compression first, and then authentication. For instance, Yu [2004] and Mokhtarian and Hefeeda [2010] proposed hash chain schemes for SVC codestreams. These schemes hash each enhancement layer and attach the hash value to the lower layer of the same frame. Recently, Zhao et al. [2012] presented an improved authentication scheme for H.264/SVC, which integrates cryptographic algorithms and Erasure Correction Codes (ECCs). They provide a mathematically provable level of security and guarantee a high security confidence. However, cryptographic-based authentication schemes are sensitive to content modifications since they cannot tell the difference between content-changing and content-preserving manipulations. Furthermore, since they must execute a hash function for each layer, their computation complexity and communication overhead are proportional to the number of layers.

Watermarking-based authentication embeds a reference object (e.g., image or message) into an SVC codestream [Shi et al. 2010; Park and Shin 2008, 2011]. Grois and Hadar [2012] provided a review of watermarking-based authentication schemes for SVC. As the reference object and the SVC codestream are mixed together, the embedded object will be tampered when the SVC codestream is maliciously tampered. For example, Meerwald and Uhl [2010] designed a robust watermarking-based authentication scheme by embedding the same watermark into both the base layer and enhancement layers for quality/spatial scalability. For the sake of robustness and security, watermarking-based authentication schemes must embed the reference object

[1]Discrete Cosine Transform (DCT) coefficients are quantized to approach zero by larger QPs; *cbp*s are zero.

into each layer of SVC; otherwise, the nonwatermarked layers can be easily tampered without being detected. However, the capacity of embedding watermarking is very limited in enhancement layers because most quantized coefficients of enhancement layers are equal to zero.

In this article, we present a novel and efficient authentication scheme for SVC surveillance codestreams, named TrustSSV (Trust Scalable Surveillance Video). TrustSSV integrates authentication and verification operations into the SVC coding process. Specifically, TrustSSV uses cryptographic authentication primitives to calculate the hash of the base layer codestream such that any bit of the base layer cannot be changed. Considering surveillance codestreams usually containing stable background scenes, TrustSSV exploits content-based authentication to protect background features and dynamical foreground features with background model support, which guarantees the integrity of enhancement layers. Our analysis indicates that the proposed scheme is secure and robust. It can allow localization of tampered regions and preserve three-dimensional SVC scalabilities. Compared with the existing authentication schemes [Wei et al. 2013, 2014a; Mokhtarian and Hefeeda 2010; Meerwald and Uhl 2010], our experimental results show that TrustSSV has very a small communication overhead and low computation complexity. It is suitable to be applied to most video streams with layered structures (e.g., HEVC/SVC, JPEG-XR).

The main contributions and key results of this article are summarized as follows:

—Novelly exploiting the multiple layer architecture (base layer and quality/spatial enhancement layers) of SVC and the temporal layers of surveillance, TrustSSV protects the integrity of surveillance video codestreams using content-based authentication and watermarking-based authentication.

—Since SVC surveillance videos have temporal scalabilities, the recipient's devices may only receive nonsuccessive frames. TrustSSV novelly performs initial background and real-time updates of background models. Once the former constructs I_b, the latter depends on I_b but has no previous frames as statistical reference to distinguish background from foreground for each AU. When there is an obvious change between background and foreground and it is constant (e.g., 5 $\sim$ 10 minutes), TrustSSV will dynamically alternate the outdated background I_b with the latest one (Update Inactive Blocks). In addition, the update of the background must be synchronous with Network Abstract Layer Units (NALUs) of the base layer; otherwise, the update of the background may be discarded with higher temporal enhancement layers.

The rest of this article is organized as follows. Section 2 presents scalable video coding. Our authentication scheme is introduced in Section 3. Experiments and analysis results are given in Section 4 and Section 5, respectively. Lastly, conclusions are drawn in Section 6.

2. SCALABLE VIDEO CODING

This section provides a quick overview of the H.264/SVC concepts and terminologies that are necessary for the understanding of the rest of the article.

2.1. H.264/SVC Standard

An SVC codestream is divided into a base layer and one or more enhancement layers, and each layer is further divided into NALUs. Due to the flexible arrangement of NALUs, SVC provides three kinds of scalabilities for the sake of bit-rate adaptation to network bandwidth and/or end devices' capabilities, as depicted in Figure 2. The three axes in Figure 2 correspond to three-dimensional scalabilities of H.264/SVC, that is, temporal, quality, and spatial scalabilities. Bars with various widths and lengths in Figure 2 refer to frames belonging to different layers. Specifically, the wider ones are at

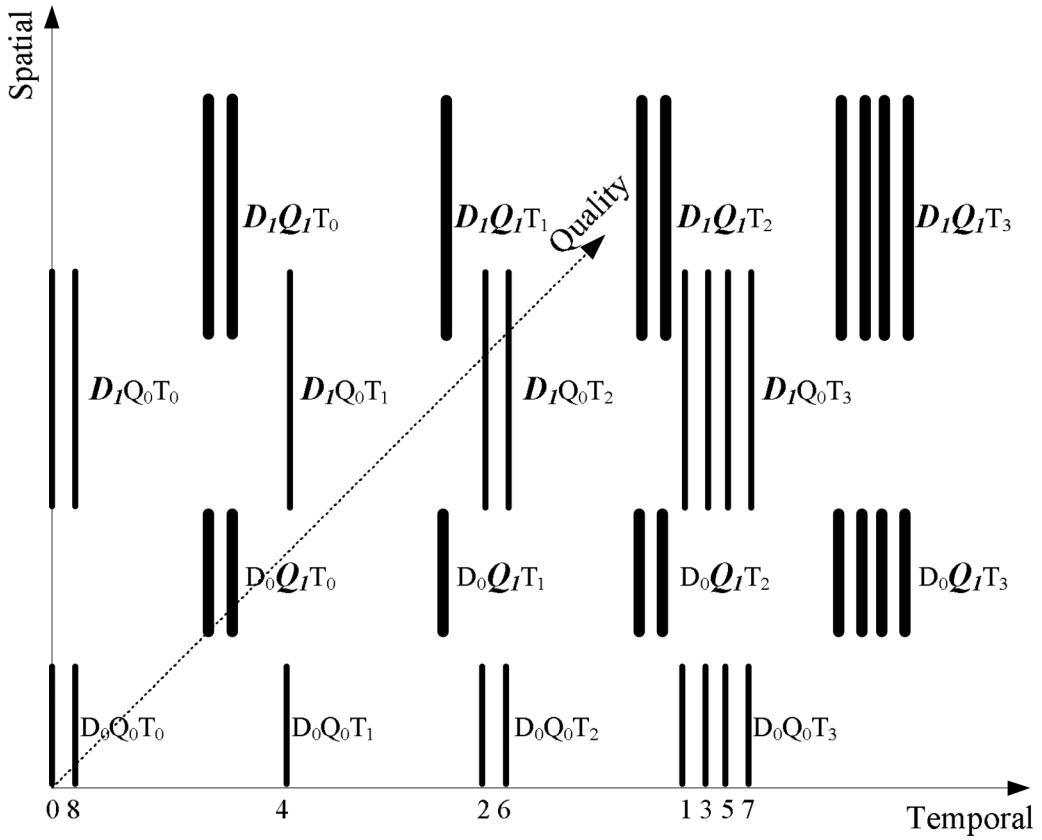


Fig. 2. An example H.264/SVC codestream with scalability in three dimensions. There are four temporal layers, two spatial layers, and one quality layer.

the quality enhancement layer, while the longer ones are at the spatial enhancement layer. In addition, $D_x Q_y T_z$ next to the bars indicates frames' spatial, quality, and temporal identification. If and only if both x and y are equal to zero, frames (bars) belong to the base layer. Otherwise, frames (bars) belong to enhancement layers. Assuming the Group of Picture (GOP) size of an H.264/SVC codestream is nine, based on the hierarchical prediction structure, nine frames will be grouped into four temporal layers as shown in Figure 2, where the numbers (0 to 8) are the order of the nine frames inside a GOP.

2.1.1. Temporal Scalability. As frames in the temporal base layers are encoded with the highest fidelity, and a lower temporal layer is used as references for motion-compensated prediction of frames in higher temporal layers, temporal scalable coding can be readily achieved; that is, by simply discarding the higher-layer frames for an SVC codestream, a lower bit-rate one is formed.

2.1.2. Spatial Scalability. The spatial base layer represents a video of the lowest resolution while the spatial enhancement layers increase the resolutions of the video. Since interlayer prediction is used, a lower spatial layer must be presented if a higher spatial layer exists, but not the other way around. Therefore, when the spatial layers are discarded starting from the highest layer, the rest of the spatial layers are still decodable.

This discarding process can be repeated until only one layer (the base layer) remains. In other words, the resolution of a video can be decreased directly and gradually.

2.1.3. Quality Scalability. The quality base layer is encoded at the lowest visual quality, and the quality enhancement layers increase the visual quality of the decoded sequence. Therefore, when the quality layers are discarded starting from the highest layer, the rest of the quality layers are still decodable. This discarding process can be repeated until only one quality layer remains.

3. AUTHENTICATION SCHEME

The present scheme seamlessly integrates authentication and verification operations into the SVC coding process. It differs fundamentally from AUSSS by utilizing a background model to detect ABlocks such that the content integrity of all ABlocks is guaranteed. We describe the background model in Section 3.1 and feature extraction in Section 3.2; the authentication module for the base layer is presented in Section 3.3, which is the same with AUSSS; and the authentication module for enhancement layers is elaborated in Section 3.4.

3.1. Background Model

Over the past years, various background models have been developed based on statistics of pixels of successive frames. Bouwmans [2011] summarizes and compares the recent advanced statistical models by classifying them into three categories. For surveillance applications, Brutzer et al. [2011] review and evaluate nine background subtraction techniques of surveillance video, and compare the performance of nine background subtraction methods with postprocessing according to their ability to meet seven challenges (e.g., gradual/sudden illumination changes, camouflage, and video noise). However, since SVC surveillance videos have temporal scalabilites, the recipient's devices may only receive nonsuccessive frames. Thus, the existing background model algorithms will result in the mismatch of detected background and foreground between providers and receivers.

In an SVC surveillance scheme, a background model should be adaptive to scene changing (e.g., switching between background and foreground) and robust to illumination changes and image processing (e.g., compression (quality scalability) or resolution resample (spatial scalability)). Based on the analysis and comparison in Brutzer et al. [2011], the mixture Gaussian model [Zivkovic and Heijden 2006] stands out among the others. In Zivkovic and Heijden [2006], each pixel is characterized by its intensity in the RGB color space. The probability of observing the current pixel value is given by

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{1}$$

where K is the number of distributions, and $\omega_{i,t}$ is a weight associated to the i^{th} Gaussian distribution at time t with mean $\mu_{i,t}$ and standard deviation $\Sigma_{i,t}$. η is a Gaussian probability density function:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t-\mu)^T \Sigma^{-1}(X_t-\mu)}, \tag{2}$$

where $X = \{X_1, \ldots, X_n\}$ are the recent histories of the color features of each pixel. In addition to the previous parameters, both the weight ω and learning rate α parameters, which are used for updating equations, need to be initialized by training in order to correctly construct a stable background model.

In order to improve the communication and computation efficiency of our authentication scheme, we first scale the original image to the same resolution as the base layer, then perform the background model. For instance, if the original sequence is Common Intermediate Format (CIF) and base layer resolution is Quarter Common Intermediate Format (QCIF), we need first to scale CIF to QCIF. The present scheme includes two background detection processing techniques. Initially, for a surveillance scene, we use the technique in Zivkovic and Heijden [2006] to generate a stable background image I_b and extract its content-based features, which represent the initial surveillance scene. Then dynamically, once I_b is constructed, different from Zivkovic and Heijden [2006], we utilize I_b but no previous frames as statistical reference to distinguish background from foreground for each AU, which solves the mismatch problem between providers and receivers due to temporal scalability. After locating foreground and background regions, the positions and content-based features of the foreground are transmitted with the AU. At the same time, we constantly detect background changes and update I_b such that I_b always represents a real-time surveillance scene.

3.2. Feature Extraction

We will first explain background operations in Section 3.2.1, then present foreground operations in Section 3.2.2.

3.2.1. Background Operations. Background operations consist of offline and online processing.

Initial background model. Assume that we obtain the background image I_b of the current SVC surveillance scene. The candidate feature extraction methods include invariant histogram statistics [Schneider and Chang 1996; Alghoniemy and Tewfik 2004; Simitopoulos et al. 2003; Kim and Lee 2003; Su et al. 2009] and the relation between low-frequency DCT coefficients [Lin and Chang 2001; Fridrich and Goljan 2000]. We utilize the NMF (Nonnegative Matrix Factorization) transform to extract content-based features of I_b, which performs the best in terms of robustness and sensitivity [Han and Chu 2010]. The NMF algorithm [Lee and Seung 2000] is able to decompose a nonnegative matrix into two nonnegative matrix factors. Monga and Mihcak [2007] exploit the NMF algorithm and propose a robust and secure image hashing method, named NMF-NMF-SQ hashing. NMF-NMF-SQ is very robust to a large class of perceptually insignificant manipulations for JPEG and is able to tolerate H.264 compression with QP = 38 [Wei et al. 2013].

Given I_b, we pseudo-randomly choose an "Inactive" Block (IB) whose resolution is $m \times m$ (e.g., $m = 4$), then perform the r_1 (e.g., $r_1 = 1$) rank NMF transform on it and output two matrices X_i and Y_i^T. The two matrices are reconnected to be a hash vector V_i^{ib} of the IB. Then, all V_i^{ib} are organized into

$$V^{ib} = \{V_1^{ib}, \cdots, V_n^{ib}\}. \tag{3}$$

Note that V^{ib} is the content-based features of I_b, where n is the total number of $m \times m$ blocks in the background image, and each component needs 12 bits. Also, note that V^{ib} is the most important and basic authentication tag; it represents the current surveillance background scene. Once a receiver needs to collect surveillance data, V^{ib} will be transmitted to each receiver over a Secure Sockets Layer (SSL) channel before sending SVC surveillance codestreams.

Real-time update of background model. Although there is a relatively stable background scene for surveillance applications, it is possible that the leaving/stopping of objects causes the background change. Specifically, the foreground switches to a new background (e.g., a motion object stops at one place for a long time), and the background

(a) Detection of foreground (b) Location of ABlocks

Fig. 3. Detection of foreground and the location of ABlocks, where white square indicates a located ABlock.

can also switch to a new background (e.g., a static object moves away for a long time). Hence, once there is an obvious change between background and foreground and the change is constant (e.g., $5 \sim 10$ minutes), the scheme will dynamically alternate the outdated background I_b with the latest one. In this article, the pixel-based distance function of the mixture Gaussian model is described as

$$D < \delta * \sigma, \tag{4}$$

where D is the substraction result, δ is the threshold of variance, and σ is the standard deviation. δ and σ can be determined by experiments.

$$D = {D_r}^2 + {D_g}^2 + {D_b}^2, \tag{5}$$

where D_r, D_g, and D_b are the distances of RGB elements. If D satisfies Equation (4), the pixel is background; otherwise, it is foreground.

A UIB (Update Inactive Block) should satisfy that it is an ABlock and keeps the same content for a long time. First, the mixture Gaussian model is based on pixel units; that is, given a pixel, it can indicate if the pixel belongs to the foreground. In this article, an $m \times m$ (e.g., $m = 4$) ABlock is constructed based on the following guidelines:

—If an $m \times m$ block does not contain foreground pixel, it is a background block.
—If an $m \times m$ block contains no less than two foreground pixels, it is an ABlock.
—If an $m \times m$ block contains only one foreground pixel and one of its eight neighboring blocks also contains one or more foreground pixels, it is an ABlock; otherwise, it is a background block.

For example, Figure 3 illustrates the detection of foreground and the location of ABlocks.

Second, the content of the ABlock is still not changing for a long time. For each ABlock, the proposed scheme further executes the NMF transform to extract content-based features, the same as with IBs. If the content-based features of an ABlock are invariant for successive frames, the ABlock in fact is a UIB.

As all UIBs with resolution $m \times m$ are detected and located, their features are organized as

$$V^{uib} = \underbrace{V_1^{uib}, \ldots, V_{n^{uib}}^{uib}}. \tag{6}$$

Their positions are organized as

$$P^{uib} = \underbrace{n^{uib}, (x_1, y_1), \ldots, (x_n, y_n)}, \tag{7}$$

where V_i^{uib} is the features of the ith UIB, n^{uib} is the number of UIBs, and $(x_i\ y_i)$ are the positions of the ith UIB. Meanwhile, the present scheme encrypts V^{uib} and P^{uib} with ciphertext (e.g., AES) in order to protect the security of the UIB against adversaries' attacks.

Although the present scheme ensures UIBs, TrustSSV does not immediately send V^{uib} and P^{uib} with NALUs of the base layer due to temporal scalability whose enhancement layers may be discarded based on receivers' requirements. Hence, the update of the background must be synchronously with the base layer NALUs whose temporal identification is zero. That is, if and only if x, y, and z of NALU scalability $(D_x Q_y T_z)$ are all zero, V^{uib} and P^{uib} can be sent along with SVC surveillance codestreams. Otherwise, the update of the background (i.e., V^{uib} and P^{uib}) may be discarded with higher temporal enhancement layers.

3.2.2. Foreground Operations. Foreground operations consist of transmission of the ABlock position and extraction of ABlock features.

Transmission of ABlock position. We make use of watermarking techniques to embed the position information of ABlocks, P, into the base layer. We adjust the last nonzero DCT coefficients of base layer blocks with odd or even.

—ABlock:
  —If its coefficients are all zero, set the last coefficient as 1.
  —If its last nonzero coefficient is odd, add 1 to the last coefficient.
  —If its last nonzero coefficient is even, do nothing.
—IB:
  —If its coefficients are all zero, do nothing.
  —If its last nonzero coefficient is odd, do nothing.
  —If its last nonzero coefficient is even, add 1 to the last coefficient.

In all, if a block has nonzero coefficients and its last coefficient is even, the block is an ABlock. Otherwise, the block is an IB. Generally, embedded watermarking may not only introduce noise to images, which decreases the visual quality of images, but also affect entropy coding and cause communication overhead. However, the proposed watermarking has less effect on visual quality and compression. First, as watermarking is embedded into the last coefficients of blocks (i.e., high-frequency band), it has little effect on vision. Second, the H.264 standard exploits motion compensation techniques to reduce temporal redundancy, and most of the IBs of enhancement temporal layers do not contain nonzero coefficients (i.e., if its coefficients are all zero, do nothing). Hence, only partial watermarking of ABlocks produces communication overhead.

Extraction of ABlock features. Assuming one frame has n^f ABlocks $\{B_i\}$, its corresponding NMF transform hash is h_i, which is concatenated from two transform matrices (the columns of X_i and the row of Y_i). Then, the hash $\mathbf{h}$ of the frame foreground is $\{h_1, h_2, \ldots, h_{n^f}\}$, where the length of h_i is v. In this article, in order to reduce the communication overhead and protect the security of extracted features, the proposed authentication scheme unitizes pseudo-random weight vectors $\{\mathbf{t}_i\}_{i=1}^{u}$ $(u \leq v)$ to compress $\mathbf{h}$, where each $\mathbf{t}_i$ is the length of v. We set u as

$$u = \begin{cases} n^f, & n^f < 16 \\ n^f/16, & n^f \geq 16. \end{cases} \tag{8}$$

The pseudo-random weight vectors $\{\mathbf{t}_i\}_{i=1}^{u}$ are generated using HC-128 [Wu 2008] with the secret key k_e and an initialization vector (IV). The initialization vector should be unique for each Access Unit (AU) such that the resulting pseudo-random vectors do

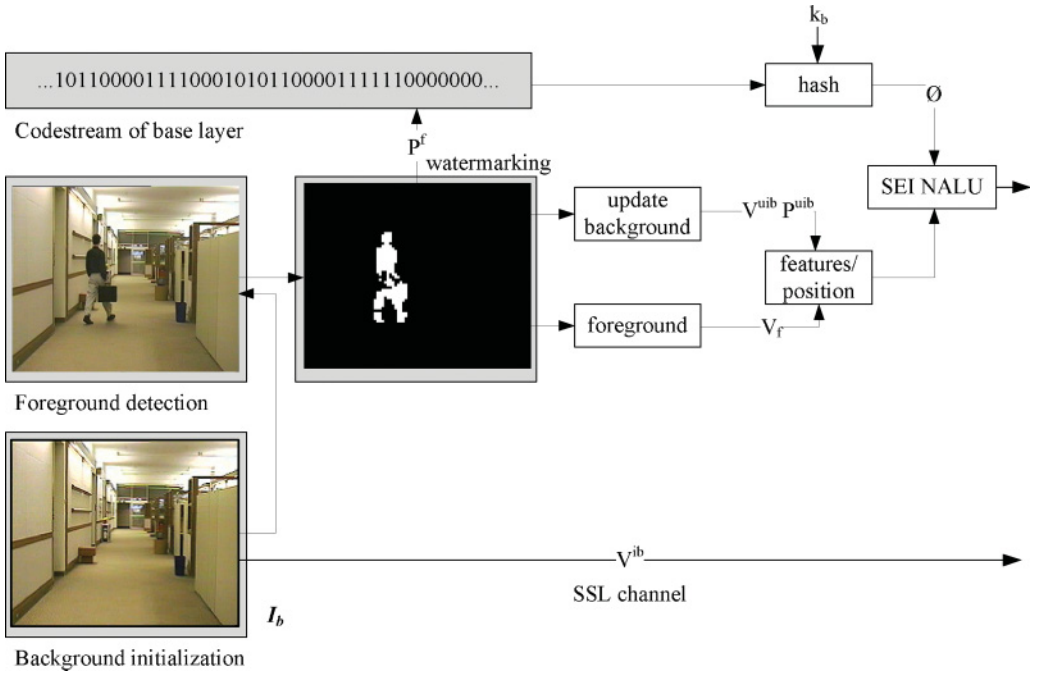


Fig. 4. Authentication flow of SVC surveillance codestreams.

not repeat themselves. In this article, IV is generated as

$$IV = \mathcal{F}(P, H_n, H_s), \tag{9}$$

where $\mathcal{F}$ is a one-way function, P is the position information, H_n represents the SVC scalable information (e.g., temporal identifier), and H_S denotes the slice header of the base layer that is protected by MAC. Because the header information is in clear text, IV can be deduced from the SVC codestream at the decoder/verifier side.

Let $V_i = \langle \mathbf{h}, \mathbf{t}_i \rangle$ be the inner product of vector $\mathbf{h}$ and vector $\mathbf{t}_i$, and then $V_f = \{V_1, \ldots, V_u\}$ is the extracted features of ABlocks $\{B_i\}$.

3.3. Authentication and Verification of Base Layer

On the provider side, the scheme takes base layer codestream Φ_b and a secret key k_b shared by provider and receiver as input to produce MAC ϕ as

$$\phi = \mathcal{H}(k_b, \Phi_b), \tag{10}$$

where $\mathcal{H}(\cdot)$ is a standard one-way hash function (e.g., SHA-1). We construct one Supplement Enhancement Information (SEI) NALU for each AU, and the calculated hash ϕ is encapsulated into its SEI NALU. The operations shown at the top of Figure 4 illustrate the authentication of the base layer.

Given a received base layer codestream Ψ_b, the receiver first calculates the MAC value ψ as

$$\psi = \mathcal{H}(k_b, \Psi_b). \tag{11}$$

Then, for the received ϕ encapsulated into the corresponding SEI NALU, if $\psi = \phi$, the authentication framework accepts the base layer's codestream. Otherwise, the codestream of the base layer is considered tampered, and both the base layer and all

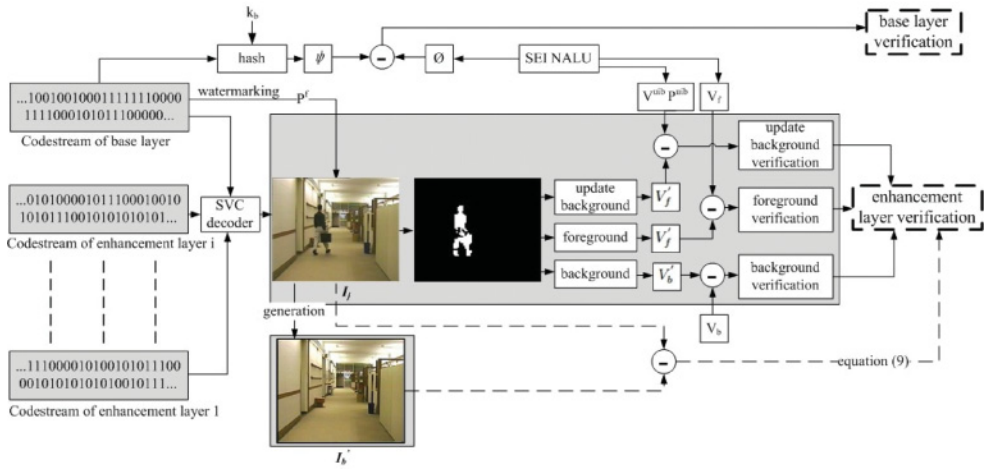


Fig. 5. Verification flow of SVC surveillance codestreams.

the enhancement layers are rejected. The top of Figure 5 illustrates the processing of base layer verification.

Moreover, temporal scalability is synchronously verified because the timestamp and frame number are authenticated by MAC such that the frame reordering attack in which the temporal order of frames may be changed can be detected.

3.4. Authentication and Verification of Enhancement Layers

On the provider side, before sending an SVC codestream, the current V^{ib} is initially sent to receivers over the SSL channel as shown in Figure 4, where V_i^{ib} is the features of the ith block ($m \times m$). During transmission of the SVC codestream, besides sending the SVC frame data, if an AU contains UIBs or ABlocks, authentication tags (V^{uib} and P^{uib}) of the updated background and authentication tags (V_f) of the foreground are also encapsulated into the SEI NALU, and then synchronously delivered along the AU.

On the receiver side, once the base layer is verified, the position information (P) of foreground ABlocks are collected from embedded watermarking. Receivers next decode the received base and enhancement layers in order to reconstruct a higher-quality/resolution image $\mathbf{I}_j$, as shown in Figure 5. Similar to the provider, the image is first scaled to the same resolution as the base layer before verifying the content integrity of the enhancement layers. The present scheme consists of foreground verification, background verification, and update background verification.

Foreground verification. Given a reconstructed image $\mathbf{I}_j$, based on position information P of the ABlocks, the receiver extracts content-based features V'_f of ABlocks as the provider does, and calculates the foreground error $e_f = \| V_f - V'_f \|$. If e_f falls within the robustness range of the content-based feature, the foreground is genuine; otherwise, it is bogus. For detailed analysis on the detection performance, including the probabilities of miss and false alarms, the reader is referred to Monga and Mihcak [2007].

Background verification. Besides ensuring the integrity of foreground ABlocks, IBs of the background also should be verified. Similar to provider sides, the present scheme also constructs a background image I'_b at receiver sides as shown in Figure 5. I'_b is originally empty.

Assume $\mathbf{I}_0$ is the first received image. Given a background block IB_i, the present scheme verifies it by extracted features $V_i^{ib'}$ and received V_i^{ib}. Similar to foreground ABlocks verification, background error $e_b^i = \| V_i^{ib} - V_i^{ib'} \|$ can decide if IB_i is modified. If IB_i is verified, it will be copied to construct the background image I_b'. Once IB_i of I_b' is filled, IB_i of following images $\mathbf{I}_j$ (j is the frame order and $j > 0$) can simply be verified based on background model parameters without performing feature extraction and content-based authentication; that is, the present scheme verifies if the received IB_i is still the background without attacks. If the pixel-based substraction distance D satisfies Equation (4), the background pixel is verified. Otherwise, a 4×4 block containing the pixel should be further verified by content-based features V^{ib} in order to decide if the block is indeed modified or there is a layer switch. SVC codestreams normally contain multiple layers; receivers may dynamically switch quality/spatial scalability during transmission. For instance, a receiver initially requires a higher-quality video of SVC codestreams and the proposed system gradually generates I_b' based on received images. Based on I_b' and Equation (4), the present scheme can quickly verify the background of the subsequent images. However, as the receiver switches the current layer to other layers, subtraction distance D may not satisfy Equation (4), which is caused by video compression. Hence, under this situation, the present scheme needs to perform content-based verification on the block in order to distinguish malicious manipulations from the layer switch operations. If the block is accepted by content-based authentication, it indicates that there is a layer switch, and I_b' will be updated by the block. Otherwise, it indicates that the block is modified, and the frame is rejected.

Update background verification. In addition, as SVC surveillance codestreams have background changing (i.e., Update Inactive Blocks), the present scheme verifies UIBs based on V^{uib} and P^{uib}. Simultaneously, outdated IBs of I_b' will be alternated by the latest verified UIBs.

Note that if and only if both the foreground and background are verified, enhancement layers are accepted; otherwise, the received enhancement layers of the frame are regarded to have been modified, and the frame is rejected.

4. EXPERIMENTS

Our experiments are carried out on a Windows 7 PC with a 2.67GHz Intel dual-core i7 processor and 4.00GB memory. We exploit Opencv[2] and FFmpeg[3] open sources to efficiently process surveillance video (e.g., capturing, scaling, or displaying); utilize JSVM 9.19 [JSVM 2011] as the encoder of SVC codestreams; take the *hall* sequence, SPEVI [SPEVI 2005], and CAVIAR Test Case Scenarios [CAVIAR 2004] as experimental datasets; link $NMFlib$[4] to extract features; and choose ***nmf_alspg*** (alternating least squares using a projected gradient method) to compute the factorization.

In our experiments, we set the GOP size and Intra period as 16, and parameters $m = 4$, $r_1 = 1$, $\delta = 5$, and $\sigma = 4$. For quality scalability experiments, the encoded SVC codestreams contain three layers (i.e., QP_{40}, QP_{35}, and QP_{20}). For spatial scalability experiments, the encoded SVC codestreams consist of one base layer (QP_{40}) and two enhancement layers (QP_{35} and QP_{20}). The QPs of enhancement layers are no more than 38 [Wei et al. 2013]. Experimental results show that the proposed TrustSSV is trustworthy because it causes low computation cost and less communication overhead and can detect tampering fields.

[2]http://opencv.org/.

[3]http://ffmpeg.zeranoe.com/.

[4]http://www.ee.columbia.edu/grindlay/code.html.

4.1. Computation Cost

As described in Section 3, the background image I_b is constructed for indoor surveillance scenes before sending SVC video data. In fact, it can be completed during setup of the surveillance system; hence, we don't count the cost (offline) in our computation complexity. We assume that I_b is generated.

Provider and receiver sides have different computation complexities. The present scheme tries to put most of the computation operations at the provider side so as to reduce receiving devices' CPU (Computer Processor Unit) and memory cost. At the provider side, authentication cost consists of base layer authentication cost A_b (i.e., an MAC operation cost) and enhancement layer authentication cost A_e. We omit A_b as compared with $A_b \ll A_e$. A_e can be represented as

$$A_e = A_e^m + A_e^{uib} + A_e^f, \tag{12}$$

where A_e^m is the background mode cost (online), A_e^{uib} is the processing cost of UIBs (e.g., feature extraction), and A_e^f is the processing cost of ABlocks (e.g., location and feature extraction of ABlock). Our experiments on sequence *motinas_room*160*_audiovisua* (360×288) indicate that A_e^m is about $0.017\mu s$ for one frame. A_e^{uib} is given by

$$A_e^{uib} = n^{uib} \cdot o(m^2 r_1), \tag{13}$$

where n^{uib} is the number of UIBs. It is the NMF transform cost that does a rank r_1 NMF on $n^{uib} \cdot m \times m$ matrices. Generally, A_e^{uib} is very small compared with the computation cost of the foreground because surveillance videos have a limited update background. A_e^f is given by

$$A_e^f = n^f \cdot o(m^2 r_1) + o(n^f m 2 r_1), \tag{14}$$

where n^f is the number of ABlocks. The first term is due to the NMF hash algorithm, which does a rank r_1 NMF on $n^f \cdot m \times m$ matrices; the second term is due to pseudo-random statistics obtained from the resulting NMF vector of length $n^f \cdot (m \times 2r_1)$.

On the other hand, the receiver's cost correspondingly contains base layer verification cost V_b and enhancement layer verification cost V_e. V_b is the same with A_b (i.e., MAC operation). V_e, however, is different from A_e: it does not have the background model cost A_e^m:

$$V_e = V_e^b + V_e^f, \tag{15}$$

where V_e^b is the verification cost of the background, and V_e^f is the verification cost of the foreground, which is the same as Equation (14). V_e^b is described as

$$V_e^b = n_1^b \cdot o(m^2 r_1) + n_2^b \cdot o\left(\sum_{i=0}^{2} (m \times m)\right), \tag{16}$$

where $n^b = n_1^b + n_2^b$ (n^b is the number of the background blocks). The first item is the content-based verification cost, and the second item is the RGB substraction verification cost. For example, our experiments on the sequence *motinas_room*160*_audiovisual* (1,068 frames) show that there are 14,471 ABlocks and no UIBs. Experimental results indicate that A_e and V_e are about $2{,}167.94\mu s$ and $3{,}910.94\mu s$ for each AU, respectively.

4.2. Communication Cost

For an SVC surveillance codestream, the present scheme should initially send the content-based features V^{ib} of the background image I_b to every receiver; hence, it causes $L^{ib} = n \times (m \times 2r_1)$ communication cost based on Equation (3). L^{ib} depends on n

Table I. Communication Overhead of Quality Scalability

Frame Name	Frame Number	ABlock (Number)	Original (Bytes)	Overhead (Bytes)	Overhead (%)
walk	1,072	11,820	2,419,725	66,492	2.75
enter	384	3,948	1,029,618	20,570	2.29
room160	1,072	18,403	6,762,669	73,076	1.08
room150	1,072	24,318	3,654,744	78,990	2.16
chamber	1,089	38,129	9,752,635	93,668	0.96
intelligent	300	2,974	851,069	18,275	2.15
hall	300	5,775	1,661,133	21,075	1.27

Table II. Communication Overhead of Spatial Scalability

Frame Name	Frame Number	ABlock (Number)	Original (Bytes)	Overhead (Bytes)	Overhead (%)
walk	1,172	5,943	2,688,493	65,715	2.44
enter	384	2,524	1,056,077	23,370	2.09
room160	1,072	8,685	6,806,130	63,357	0.93
room150	1,075	10,512	3,708,985	65,337	1.76
chamber	1,120	18,096	10,134,994	75,216	0.72
intelligent	300	2,443	858,499	17,743.5	2.07
hall	300	2,934	1,583,646	18,234	1.15

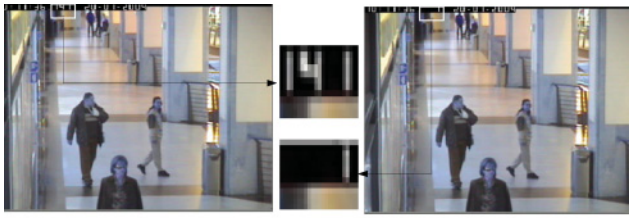
and r_1. For instance, assume the resolution of I_b is QCIF, n is equal to $(\frac{176}{m} \times \frac{144}{m} = 1{,}584)$, and L^{ib} has 12,672 hash elements. Since V^{ib} is delivered to receivers before sending SVC data under the SSL channel, we do not count them into our communication cost in this article.

Besides V^{ib}, the present scheme needs to utilize extra SEI NALUs to carry the MAC value of the base layer, the position and features of UIBs (P^{uib} and V^{uib}), and content-based features of ABlocks (V_f) for each AU. In all, the communication cost is given by

$$L = L^{sei} + L^b + L^f + L^{uib}, \tag{17}$$

where L^{sei} is the length of the SEI header, L^b is the MAC value length of the base layer, L^f is the overhead caused by the foreground, and L^{uib} is the overhead produced from the update of the background. L^{uib} is variable; while L^{sei} and L^b are constant, they are equal to 19 and 32, respectively. L^f depends on u; for example, u is equal to 16, and the features length of the foreground is equal 24 bytes.

The experimental results on the test sequences are shown in Table I and Table II. *walk* and *enter* refer to $walkbyshop1front$ and $enterexitcrossingpaths1front$, respectively. *room*160 and *room*150 refer to $motinas\_room160\_audiovisual$ and $motinas\_room150\_audiovisual$, respectively. *chamber*, *intelligent*, and *hall* refer to $motinas\_chamber\_audiovisual$, $intelligentroom\_raw$, and *hall*, respectively. *walk* describes a couple walking along corridor browsing and people going inside and coming out of stores. *enter* describes two people crossing paths at the entrance of a store and a couple walking on the corridor. *room*160 and *room*105 are recorded in rooms with reverberations, while *chamber* is recorded in a room with reduced reverberations. *intelligent* records a student who does some activities with a static camera, while *hall* records two people walking along a corridor. The average communication overheads are 1.81% and 1.60% for quality and spatial scalability, respectively.



(a) Original image. (b) Attacked image.

Fig. 6. Removing attack of foreground.

(a) Original image. (b) Attacked image.

Fig. 7. Replacing attack of foreground.

4.3. Tampering Detection

In this subsection, tampering detection experiments are performed on protected SVC surveillance codestreams, and detected results are illustrated and analyzed. Since cryptographic authentication (e.g., MAC) is performed on the base layer, adversaries cannot attack any bits of the base layer. Hence, we only describe the tampering experiments of enhancement layers, for example, basic attacks (color and illuminate manipulations) or object attacks (inserting, removing, and replacing manipulations). We will discuss foreground and background attacks, respectively.

Foreground tampering detection. Foreground ABlocks are located by the watermarking of the base layer. It is impossible for adversaries to modify foreground content without the secret key k_e because once the content of the foreground is modified (e.g., replacing, removing, or inserting), the extracted features V'_f will not be verified by V_f. Figure 6 illustrates that original time (141) is removed. Figure 7 illustrates that the man's face is replaced by a woman's face.

Background tampering detection. Based on Section 3.4, there are two verification processing techniques. First, if the IBs of I'_b are not filled, we make use of content-based features to verify IBs. If a background IB is attacked, its content-based features mismatch from the original one, and then the tampering block is detected. Second, after the generation of background image I'_b, the present scheme verifies enhancement layers by the subtraction operations. For each background block, the present scheme calculates the pixel-based (RGB) distance based on Equation (4). If the distance is smaller than the threshold, the pixel is accepted. Otherwise, the block containing the pixel should be verified by content-based authentication in order to decide if the unsatisfied distance is caused by malicious manipulations or layer switch operations. Generally, meaningful content attacks must cause large distance because they change the texture, luminance, or color of attacked blocks. For example, Figure 8 illustrates an inserting attack of the background; a new rubbish bin is inserted into the background. Figure 9 illustrates a removing attack of the background; a small object at the bottom

(a) Original image (b) Attacked image

Fig. 8. Inserting attack of background.

(a) Original image (b) Attacked image

Fig. 9. Removing attack of background.

of the door is removed. Both content-based and substraction verifications can detect the attacks and locate the tampering places.

5. PERFORMANCE ANALYSIS

5.1. Security

Given the input k_b and Φ_b, the present scheme outputs an MAC that ensures that a receiver who knows the secret key k_b can detect any changes to the base layer. Hence, it is impossible that adversaries modify the content of the base layer without being detected.

Besides the base layer, SVC codestreams also supply various enhancement layers in order to produce higher-quality/resolution images. For the background, we assume that the content-based features of the initial background image are transmitted to receivers over an SSL channel, which ensures the security of V^{ib}. For an updated background, the position information P and content-based features V^{uib} of UIBs are encrypted by a block cipher such as AES. It's well known that block ciphers can be regarded as secure pseudo-random permutations, and it is computationally infeasible to distinguish the output of a block cipher from that of a truly random permutation [Katz and Lindell 2008]. For the foreground, since the content-based features V_f of ABlocks are the inner product of NMF hash $\boldsymbol{h}$ and pseudo-random weight vector $\{\mathbf{t}_i\}_{i=1}^{u}$, this implies that an attacker cannot forge V_f without the knowledge of the secret key k_e, although the attacker knows the content of SVC codestreams.

5.2. Scalability

The present scheme preserves the scalability of the original SVC codestreams because the proposed authentication scheme does not affect the standard structure of SVC codestreams. That is, MANEs (Media Aware Network Elements) can still transparently adapt SVC codestreams based on receivers' scalable requirements. As receivers only

Table III. Comparison with Other H.264/SVC Authentication Schemes, QS (Quality Scalability) and SS (Spatial Scalability)

Scheme	Tampered Location	Authentication Operation	Dependence on SVC Structure	Communication Overhead
Cryptographic authentication [Mokhtarian and Hefeeda 2010]	No	All layer hash	Yes	QS 2.19% SS 2.62%
Watermarking authentication [Meerwald and Uhl 2010]	Yes	All layer watermarking	Yes	QS 8.66% SS 2.93%
Proposed authentication	Yes	Base layer hash content-based features	No	QS 1.81% SS 1.60%

obtain the base layer codestream, the proposed scheme verifies it by cryptographic-based authentication (i.e., MAC). As receivers obtain higher-quality and/or spatial enhancement layers, the proposed scheme further verifies them by content-based authentication, that is, authenticating once, verifying many ways [Wu and Deng 2006].

5.3. Robustness and Sensitivity

The base layer is protected by cryptographic-based authentication; thus, the present scheme is sensitive to any bits changing for the base layer. This is suitable for SVC applications as discussed in Section 1. That is, the base layer is the basic and the most important component of SVC codestreams such that it must be transmitted to clients at any session. In addition, since the base layer as the previewing version contains the lowest-quality and -resolution images, there is no need to perform transcoding on the base layer. Hence, it is reasonable that the base layer is sensitive to any bit changing.

For enhancement layers, the present scheme employs the content-based authentication to ensure their integrity. The robustness of content-based features is suitable for scalable properties of SVC codestreams, that is, authenticating once, verifying many ways. For example, a receiver may obtain different quality/resolution images under various network or device requirements, but all of them can be verified by their content-based features. On the other hand, the sensitivity of content-based features can distinguish incidental manipulations from malicious manipulations. Hence, attacked enhancement layers will be detected and rejected as described in Section 4.3.

5.4. Comparison with Other Schemes

Compared with the hybrid scheme [Wei et al. 2014a], the communication overhead of the proposed scheme is 1.81% and 1.60% for quality scalability and spatial scalability, respectively, which is less than 2.42% and 2.16% of the hybrid scheme. Compared with the AUSSS scheme [Wei et al. 2013], the scheme proposed here causes similar communications overhead with Wei et al. [2013] and overcomes the shortcoming of Wei et al. [2013], which may bypass ABlocks when an SVC codestream contains a lower-quality base layer.

Table III describes the performance of the proposed TrustSSV as compared with existing cryptographic and watermarking authentication schemes.

—**Tampered location:** cryptographic authentication cannot locate tampered locations, while the present scheme and watermarking scheme can.

—**Authentication operation:** TrustSSV only depends on base layer codestreams and content-based features of SVC surveillance data, while cryptographic authentication and watermarking-based authentication must involve hash or watermarking of every SVC layer to prevent the attacks on unprotected layers. Hence, TrustSSV produces constant communication overhead, but cryptographic authentication [Mokhtarian

and Hefeeda 2010] and watermarking-based authentication [Meerwald and Uhl 2010] are variable. Their overhead increases with the number of enhancement layers. For example, with a GOP size of 8, each frame will carry 40 bytes more overhead [Mokhtarian and Hefeeda 2010] when an SVC sequence contains one more enhancement layer.

—**Dependence:** cryptographic authentication and watermarking-based authentication depend on a layer prediction relationship of SVC in order to construct hashing chain or embed watermarking, while TrustSSV is independent of the SVC structure; that is, it is transparent to users.

—**Communication overhead:** suppose an SVC codestream with three spatial/quality layers and a GOP size of 16. Compared to the state-of-the-art methods (i.e., cryptographic and watermarking authentication), the cryptographic authentication scheme in Mokhtarian and Hefeeda [2010] incurs a communication overhead of around 2.19% and 2.62% of the GOP size for quality and spatial scalability, respectively, whereas the watermarking authentication scheme in Meerwald and Uhl [2010] incurs around 8.66% and 2.93% communication overhead for quality and spatial scalability, respectively.

6. CONCLUSION

In this article, a secure and robust authentication scheme was proposed for SVC surveillance codestreams. According to the quality/spatial layer characteristics of SVC codestreams, the proposed scheme exploited cryptographic-based authentication to protect the base layer and content-based authentication to ensure the content integrity of quality/resolution enhancement layers. Based on the fact that most surveillance videos contain stable scenes, in order to increase the efficiency of authentication/verification, the proposed scheme utilized a background model to differentiate background from foreground and initially send background features once and dynamically update foreground features in real time. Evaluation indicates that TrustSSV preserves the SVC scalability property, detects tampered regions, and can be applied to most video streams with layer structures (e.g., HEVC/SVC, JPEG-XR). Experimental results showed that TrustSSV introduces much lower communication overhead than cryptographic-based and watermarking-based authentication, and has low computation complexity. Therefore, TrustSSV is trustworthy and suitable to big video data applications.

REFERENCES

M. Alghoniemy and A. H. Tewfik. 2004. Geometric invariance in image watermarking. *IEEE Transactions on Image Processing* 13, 2 (2004), 145–153.

T. Bouwmans. 2011. Recent advanced statistical background modeling for foreground detection: A systematic survey. *Recent Patents on Computer Science* 4, 3 (2011), 147–176.

S. Brutzer, B. Höferlin, and G. Heidemann. 2011. Evaluation of background subtraction techniques for video surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition* (2011), 1937–1944.

CAVIAR. 2004. CAVIAR test case scenarios. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1 (2004).

J. Fridrich and M. Goljan. 2000. Robust hash function for digital watermarking. *Processing IEEE International Conference on Information Technology: Coding Computing* 11, 2 (2000), 178–183.

D. Grois and O. Hadar. 2012. Recent advances in watermarking for scalable video coding. *Watermarking, Intech Open Access Publisher* (2012).

S. H. Han and C. H. Chu. 2010. Content-based image authentication: Current status, issues, and challenges. *International Journal of Information Security* 9, 1 (2010), 19–32.

M. Hefeeda and K. Mokhtarian. 2011. Authentication of scalable multimedia streams. In *Handbook on Security and Networks*, Y. Xiao, F. H. Li, and H. Chen, Eds. World Scientific Publishing Co. (2011), 93–125.

JSVM. 2011. Joint scalable video model software. http://ip.hhi.de/imagecomg1/savce/downloads/svc-reference-software.htm. (2011).

J. Katz and Y. Lindell. 2008. *Introduction to Modern Cryptography*. Chapman & Hall/CRC (2008).

H. S. Kim and H. K. Lee. 2003. Invariant image watermark using zernike moments. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 8 (2003), 766–775.

D. D. Lee and H. S. Seung. 2000. Algorithms for non-negative matrix factorization. In *Proceedings of the 2000 Conference on Advances in Neural Information Processing Systems 13*. MIT Press (2000), 556–562.

C. Y. Lin and S. F. Chang. 2001. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and System for Video Technology* 11, 2 (2001), 153–168.

P. Meerwald and A. Uhl. 2010. Robust watermarking of H.264/SVC-encoded video: Quality and resolution scalability. In *International Workshop on Digital Watermarking* (2010), 156–169.

K. Mokhtarian and M. Hefeeda. 2010. Authentication of scalable video streams with low communication overhead. *IEEE Transactions on Multimedia* 12, 7 (2010), 730–742.

V. Monga and M. K. Mihcak. 2007. Robust and secure image hashing via non-negative matrix factorizations. *IEEE Transactions on Information Forensics and Security* 2, 3 (2007), 376–390.

S. W. Park and S. U. Shin. 2008. Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC). In *New Directions in Intelligent Interactive Multimedia, Springer, Studies in Computational Intelligence* (2008), 351–361.

S. W. Park and S. U. Shin. 2011. Authentication and copyright protection scheme for H.264/AVC and SVC. *Journal of Information Science and Engineering* 27, 1 (2011), 129–142.

M. Schneider and S. F. Chang. 1996. A robust content based digital signature for image authentication. In *International Conference on Image Processing* (1996), 227–230.

H. Schwarz, D. Marpe, and T. Wiegand. 2007. Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and System for Video Technology* 17, 9 (2007), 1103–1120.

F. Shi, S. H. Liu, H. X. Yao, Y. Liu, and S. P. Zhang. 2010. Scalable and credible video watermarking towards scalable video coding. In *Pacific-Rim Conference on Multimedia* (2010), 697–708.

D. Simitopoulos, D. E. Koutsonanos, and M. G. Strintzis. 2003. Robust image watermarking based on generalized radon transformations. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 8 (2003), 732–745.

SPEVI. 2005. Surveillance performance evaluation initiative (SPEVI) datasets. http://www.elec.qmul.ac.uk/staffinfo/andrea/spevi.html (2005).

P. C. Su, C. C. Chen, and H. M. Chang. 2009. Towards effective content authentication for digital videos by employing feature extraction and quantization. *IEEE Transactions on Circuits and Systems for Video Technology* 19, 5 (2009), 668–677.

L. Wei, H. Zhu, Z Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos. 2014b. Security and privacy for storage and computation in cloud computing. *Information Sciences* 258 (2014), 371–386.

L. Wei, H. Zhu, Z Cao, W. Jia, and A. V. Vasilakos. 2010. SecCloud: Bridging secure storage and computation in cloud. In *IEEE 30th International Conference on Distributed Computing Systems Workshops* (2010), 52–61.

Z. Wei, R. H. Deng, J. L. Shen, Y. D. Wu, X. H. Ding, and S. W. Lo. 2013. Technique for authenticating H.264/SVC bit streams in video surveillance applications. In *IEEE International Conference on Multimedia and Expo* (2013).

Z. Wei, X. H. Ding, Robert H. Deng, and Y. D. Wu. 2012. No tradeoff between confidentiality and performance: An analysis on H.264/SVC partial encryption. *Communications and Multimedia Security* (2012), 72–86.

Z. Wei, Y. D. Wu, Robert H. Deng, and X. H. Ding. 2014a. A hybrid scheme for authenticating scalable video codestreams. *IEEE Transactions on Information Forensics and Security* 9, 4 (2014), 543–553.

H. Wu. 2008. The stream cipher HC-128. *New Stream Cipher Designs* (2008), 39–47.

Y. D. Wu and R. H. Deng. 2006. Scalable authentication of MPEG-4 streams. *IEEE Transactions on Multimedia* 8, 1 (2006), 152–161.

H. Yu. 2004. Scalable streaming media authentication. In *IEEE International Conference on Communications* (2004), 1912–1916.

Y. F. Zhao, S. W. Lo, R. H. Deng, and Xuhua Ding. 2012. An improve authentication scheme for H.264/SVC and its performance evaluation over non-stationary wireless mobile networks. *Journal of Computer and System Sciences* 7645 (2012), 192–205.

B. B. Zhu, M. D. Swanson, and S. Li. 2004. Encryption and authentication for scalable multimedia: Current state of the art and challenges. In *Proceedings SPIE International Symposium Information Technology & Communication* (2004), 157–170.

Z. Zivkovic and F. Heijden. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters* 27 (2006), 773–780.