A Novel Stream Video Integrity Method

CHEN Jianmin
1. College of Computer Science, Beijing University of Technology, Beijing, 100124, China
2. National Computer Virus Emergency Response Center, Tianjin 300457, China
cjm@antivirus-china.org.cn

ZOU Shihong
3. Beijing University of Posts & Telecommunications, Beijing, 100876, China

REN Aihong
4. School of Computer Science and Technology, Xidian University, Xi'an, 710071, China
5. Department of Mathematics, Baoji University of Arts and Sciences, Baoji, 721013, China)

Abstract—**The paper analyses the traditional methods of the stream video integrity technology and gives some possible signature schemes for video integrity, including the batch signature which can improve the efficiency in signature generation, sanitizable signature which can tolerate non-malicious operation, and Merkle-tree signature. What's more, we present a new idea for video integrity based on the batch signature scheme, which is more efficient than traditional methods.**

Index Terms—**digital signature, video integrity, batch signature, sanitizable signature**

I. INTRODUCTION

The well-known adage that "seeing is believing" is no longer true due to the pervasive and powerful multimedia manipulation tools. Such development has decreased the credibility that multimedia data such as photos, video or audio clips, printed documents, etc. used to command. To ensure trustworthiness, multimedia authentication technique is being developed to protect multimedia data by verifying the information integrity, the alleged source of data, and the reality of data. The digital watermarking and digital signature are two techniques used to address this issue. We just focus on the signature for video integrity in this paper.

Digital watermark techniques embed an invisible signal (for example, company logo or personal symbol) into video so as to attest the owner identification of the media and discourage the unauthorized copying. While watermark techniques emphasize protecting the right of service providers, digital signature focuses on that of the customers. For example, a video purchaser may want to know whether the product he or she bought is from the legal seller and is the authentic one. Digital signature scheme can be used to solve this problem. First the video seller extracts some information dependent on the content of the original video and encrypts it into a small-size file, which is called signature. Then the signature file is sent to the purchaser with the original video[1]. An obvious drawback of these schemes is the extra bandwidth needed for transmission of the signature. Because most digital applications such as Internet multimedia, wireless video, personal video recorders, video-on-demand, videophone and videoconference have a demand for much higher compression to meet bandwidth criteria and best video quality as possible, different video codecs have evolved to meet the current requirements of video application based products. Among various available standards, H.264/AVC Advanced Video Codec is becoming an important alternative providing reduced bandwidth, better image quality in terms of peak-signal-to-noise-ratio (PSNR) and network friendliness[2], but it requires higher computational complexity.

The paper is organized as following: in section 2, we analyse the current methods for video integrity technology, in section 3, we give three possible schemes for video signature, include the batch signature, sanitizable signature and merkle-tree signature. The batch signature is more efficient than the common signature in generating signature, the sanitizable signature can satisfy that the censor may modify signed document or video without interation with the signer (in limited and controlled fashion). In section 4, give our new idea. Finally, we give the conclusion in section 5.

II. ANALYSIS FOR TRHANDITIONAL METHODS

A large number of watermarking schemes have been proposed for copyright protection and authentication for current popular standards such as MPEG-1 and MPEG-2, but only a few for the latest video coding standard H.264/AVC. In addition, as many new features are introduced to H.264, a large number of previous video watermarking algorithms cannot be applied directly, so development of new algorithms is required to address this new standard.

The state-of-the-art watermarking research and technology to authenticate the H.264/AVC video falls into two broad classes: digital watermarking and digital signature. Digital watermarking directly embeds some

doi:10.4304/jcp.8.7.1861-1866

information into video. Some of the published H.264/AVC video authentication papers have concentrated on embedding a watermark directly in the compressed domain[3-4]. In a few others, the embedding process is carried out in the compressed bit-stream delivered by the H.264/AVC encoder[5-7]. Until now, most of the compressed-domain (during encoding) video authentication systems for H.264/AVC takes into account the temporal dimension of the video and rely on marking the motion vector. In [4], the authors proposed a hard authentication algorithm to authenticate the H.264/AVC video based on the accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization for mode decision of the H.264/AVC. The algorithm performed well in terms of sensitivity against transcoding and common signal processing but lacked the ability to provide further information necessary to characterize the attack. Digital signature is a conventional scheme used in [8] to authenticate the H.264/AVC. The digital signature is embedded as Supplemental Enhancement Information (SEI) in the H.264/AVC bit-stream. The drawback of their scheme is the increase in the bits transmitted by the encoder, so the extra bandwidth needed for transmission of video.

To address the problem of the extra bandwidth needed for transmitting the signature, a combined digital watermark and digital signature for compressed H.264/AVC video authentication and content integrity verification is proposed in [9]. The digital signature treated as a fragile watermark is generated from the video contents and then inserted into H.264/AVC stream during encoding process. The watermark is embedded by selecting suitable motion vectors (MVs) which are associated with higher motion activities within P-frames by forcing their Least Significant Bits (LSB) to match the corresponding watermark bits. To authenticate and verify the received compressed video, the receiver performs the same operations as applied on the embedding side in a reversed order to extract the embedded watermark and compares it with the signature generated in the decoder in the same manner as that employed by the embedder. If the signature and the extracted watermark match, the received video is considered to be authentic. Otherwise, the embedded watermark will degrade the original video, which makes the signature extracted from the watermarked video different from the original one. Therefore, a robust feature extraction to generate the signature is of great importance.

There are two types of robust digital signature for video. The first type generate digital signature based on the pixel values of each picture (refer to figure 1). The second type can generate the digital signature picture by picture (refer to figure 2).

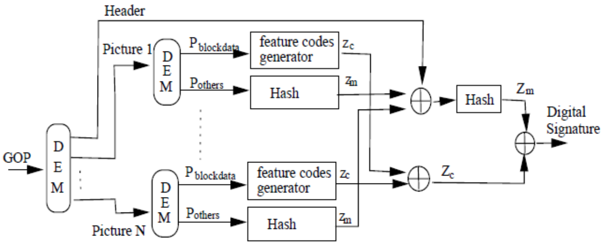

Figure 1 Robust digital signature: type 1

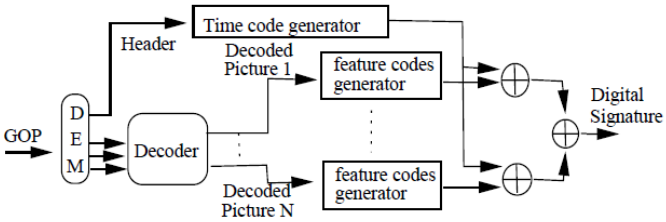

Figure 2 Robust digital signature: type 2

These two types generating digital signature also used the common signature method. Except the common method, there is some others method, for example, paper [10], which is sensitive to spatial and temporal tampering, and also robust to frame dropping. This method is suitable for the scenario of video streaming through a communication channel. Due to the large size of video data, the video streaming often suffers from congestion problem at the bottlenecks on the network. To overcome the network congestion problem, some data loss (for example, loss of few video frames) is inevitable. The authors exploit cryptographic secret sharing and the temporal relationship in video to afford frame drops yet maintain the integrity of the video. The core idea of this technique is to utilize three hierarchical levels of a video and to use cryptographic secret sharing to create what we call as a "secret frame". The authors authenticate a given video by computing the secret frames based on randomly generated private keys at three hierarchical levels i.e. key frame level, shot level, and video level. Firstly, segment the video into shots. Then, for each shot we identify the key frames. At the key frame level, compute the secret for each pair of key frames using secret sharing considering all non-key frames between the two key frames as shares. The secrets computed at this level and the key frames are treated as shares to compute the secret at the shot level. Finally, all shot secrets are used to compute a master secret that is considered as the signature for the video. In this scheme, the size of the authenticating signature is equal to a video frame size irrespective of the length of the video in time.

III.Some Possible signature for Video Integrity

In the surveillance video system, the signers are usually cameras which have limited resources and the verifiers have unlimited resources to check the messages. The currently method is not practical because of the efficiency for the surveillance video system. So we must improve the efficiency for the generating signature in surveillance video system. Now we introduce some possible signatures for video integrity.

3.1 Batch Signature

Generate digital signatures for many messages have high computational load, which typically require modular exponentiation. Some researchers specifically address this inefficiency by introducing a new signature generation scheme ------ batch signature, that is able to sign many messages for almost the cost of one signature operation at the serve, or signing, entity.[11-12]

A batch signature consists of an ordinary signature which depends on every message in the batch, and a batch residue which varies with every message (the batch residue is a component of the batch signature on the message). Signature verification consists of recalculating the input to the ordinary signature using the message and batch residue, and then verifying the ordinary signature. Because calculation of the batch residue, ant its verification, only use hash calculations. The generation of the batch signature is almost as efficient as generation of a single ordinary signature. But the batch signatures are longer than ordinary signatures because of the need to accommodate a batch residue. In table 1, we analyze the efficiency of different batch signatures and common signature.

TABLE 1
Efficiency analysis for batch signature

	sign	Ver.	Size of batch residue	
			Ave. size	Complexity
Ordinary Sig.	1exp.+1hash	1exp.+1hash	-	-
Simple batch Sig.	1exp.+ (b+1)hash	1exp.+2hash	$(b+1)\|h\|$	$O(b)$
Tree-based batch Sig.	1exp.+(1+$2^{k+1}\cdot$ (k+1))hash	1exp.+2hash	$(k+2r/(2^k+r))(\|h\|+1)$	$O(\log b)$

The parameter is as following: $|h|$ means the size of hash value, $b=2^k+r$ means the number of batch message, where $r<2^k$.

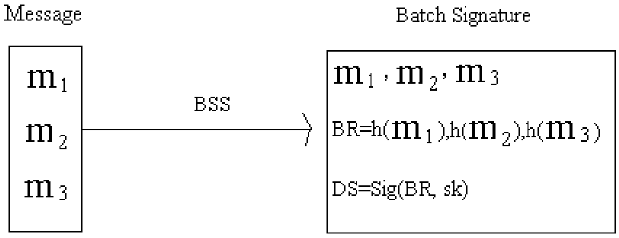

Figure 3 Simple batch signatures

There are two kinds of batch signature. The first one is simple batch signature (figure 3); the second one is tree-based batch signature (figure 4). In general, an m-ary tree can be used instead of a binary tree to reduce the number of additional nodes needed, but the message size overhead is higher.

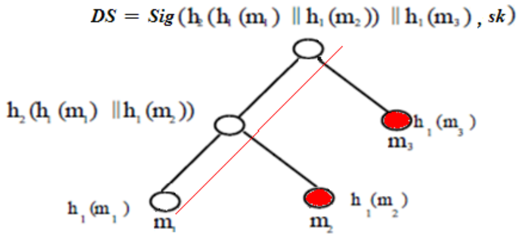

Figure 4 Tree-based batch signatures

In the surveillance video system, the signers are usually cameras which have limited resources and the verifiers have unlimited resources to check the messages. But the batch verification signature conflict with the requirements of surveillance video system, and at the same time, the batch signature is accord with the requirement. So we think the tree-based batch signature is a good candidate for the surveillance video system. But there have a challenge that is minimize the size of batch residue. In next plan, we well try to improve this scheme from the size of batch residue.

3.2 Sanitizable Signature

If someone needs to refer to a sanitized document, it is necessary to ascertain the source and the integrity of the document in order to avoid liability. The technique of plain digital signature (e.g., RSA or ECC) can achieve both source ID authentication and data integrity. More exotic constructs, such as redactable signatures[13], allow anyone to obtain a valid signature of the redacted document without any help from the original signer. However, there are situations where a duly authorized third party may need to modify the document in some controlled and limited fashion. The authorized third party needs to somehow come up with a valid signature for the updated document, without contacting the original signer. Many possible reasons for not asking the original signer to re-sign, including: (1) the signer's key has expired, (2) the original signature was securely timestamped via, e.g., [14], (3) the signer may not be reachable/available, (4) each new signature would cost too much, either in terms of real expense or in terms of computation. In this paper, we introduce the notion of sanitizable signatures precisely in order to address these needs.

Informally, a sanitizable signature scheme allows a semi-trusted third party to modify designated portions of the document and produce a valid signature on the legitimately modified document without any help from the original signer. These designated portions of the document are blocks or segments explicitly indicated as mutable under prior agreement between the signer and the censor. The third party can produce a valid signature only if it modifies these portions and no other parts of the message. (Refer to the figure 5.)

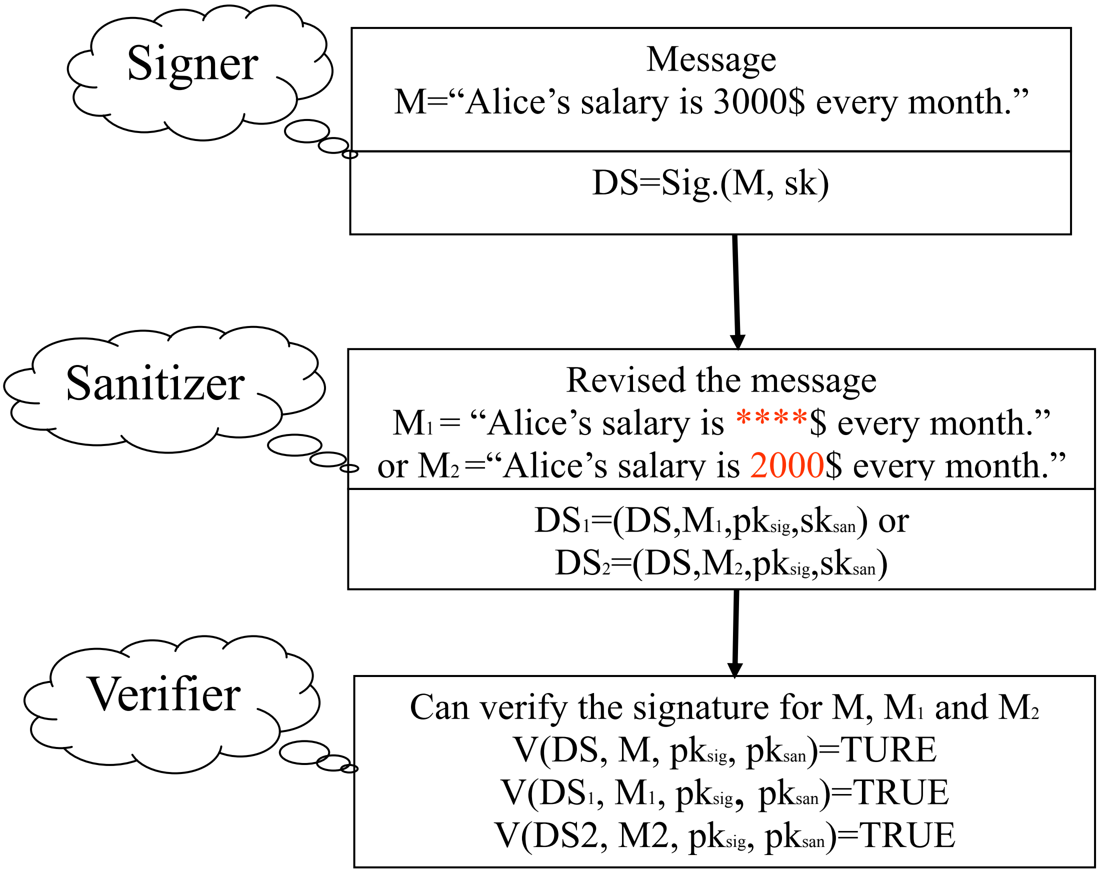
Signer
Message
M=“Alice’s salary is 3000\$ every month.”
DS=Sig.(M, sk)
Sanitizer
Revised the message
M_1 = “Alice’s salary is ****\$ every month.”
or M_2 =“Alice’s salary is 2000\$ every month.”
$DS_1=(DS,M_1,pk_{sig},sk_{san})$ or
$DS_2=(DS,M_2,pk_{sig},sk_{san})$
Verifier
Can verify the signature for M, M_1 and M_2
$V(DS, M, pk_{sig}, pk_{san})$=TURE
$V(DS_1, M_1, pk_{sig}, pk_{san})$=TRUE
$V(DS2, M2, pk_{sig}, pk_{san})$=TRUE

Figure 5 Sanitizable signature

There are some previously proposed schemes. Miyazaki et al. [15] proposed the schemes called SUMI-1, SUMI-2, SUMI-3, and SUMI-4. In the model of the schemes, the sanitizer can sanitize any message he wants. The signer cannot restrict sanitization. Steinfeld, Bull, and Zheng[16] proposed CES-CV, CES-HT, CES-RSAP, and CES-MERP. They are provably secure. In the model of the schemes, the signer can assign each message whether it can be sanitized or not. However, the signer cannot change his assignment once the signer generates the signature. Miyazaki et al. [17] proposed SUMI-5. In the model of the scheme, the signer can change his assignment even after he generates the signature. Miyazaki et al. [18] also proposed SUMI-6. Miyazaki et al. [18] claims that the scheme can hide the number of sanitized messages of the document. Ateniese, Chou, Medeiros, and Tsudik introduced the sanitizable signatures[19]. In the model of the scheme, only the designated sanitizer can sanitize the document. Even the signer cannot sanitize the document after he generates the signature. Notice that the meaning of ‘sanitize’ in this scheme is different from the other protocols. In this scheme, ‘sanitize the message’ means ‘change the message’, while the other schemes, ‘sanitize the message’ means ‘hide the message’.

We think the ideas of sanitizable signature can be used in surveillance video, but we cannot use it directly because of the low efficiency.

3.3 Merkle-tree Signature

In 1979 Ralph Merkle proposed a new signature scheme ------ Merle-tree signature which can avoid this situation that each key pair can only be used for one signature in one-time signature[20]. His idea is to use a complete binary hash tree to reduce the validity of an arbitrary but fixed number of one-time verification keys to the validity of one single public key, the root of the hash tree.

We verify the signature using the one-way function to the signature values and comparing them with the corresponding public key. If the values are equal, we think the signature is valid, otherwise invalid. The security of the scheme relies on the one-way property of the hash function. When we verify the signature, we reveal a part of the secret key. Therefore we can use the given secret/public key only once.

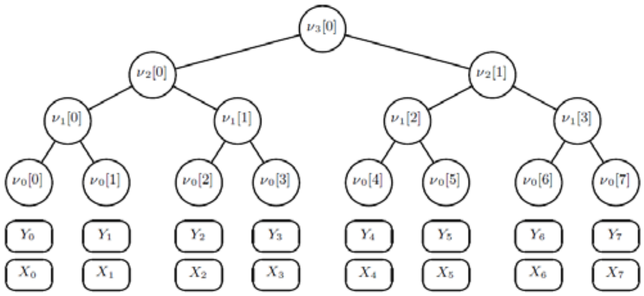

Figure 6. A Merkle-tree of height H=3

We can know the Merkle-tree signature (figure 6) is more efficient than others signature, so we think it is a good candidate for video signature.

IV. NEW DATA INTEGRITY METHOD FOR THE VIDEO

Multimedia compression standards have been designed and widely adopted by various applications: JPEG in the WWW, MPEG-1 in VCD, MPEG-2 format in DVD, and H.261 and H.263 in video conferencing. The source of a multimedia authentication system may be raw data or compressed data. In practical applications, the raw format of multimedia data may not be available. For instance, a scanner generates temporary raw images but only saves them in their compressed format; a digital camera which captures image/video produces compressed files only, without generating any raw data. Therefore, an authentication system which can only authenticate raw data may have limited uses in practice. So we just consider the compressed data.

From the figure 7, we can see that the signature systems is not only low efficiency, but also large size of the signature, so it not suitable for the stream video. The Merkle-tree signature is more efficient, but there have a challenge that is minimize the size of batch residue. In order to solute this problem, we give a new idea for the batch signature, which can reduce the size of batch residue.

In our improved batch signature, we distribute the message not only the leaf node, but also the root node (figure 8).

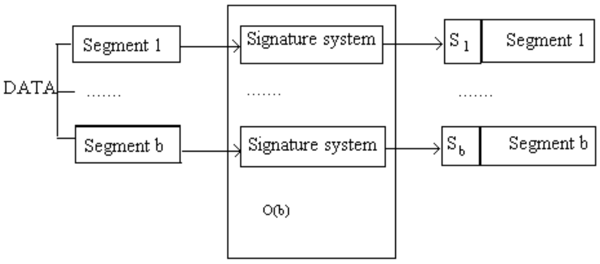

Figure 7. Current signature system

1) Key generation procedure

For security parameter 1^m, we generate *k* random strings, each of *m* bits, to produce the secret key:

$$s_k = (s_1, s_2,.., s_k).$$

The corresponding public key is:

$$p_k = (v_1, v_2,.., v_k) = (h(s_1), h(s_1),..., h(s_1)),$$

Where h is a one-way function operating on m-bit strings.

2) Signature generation procedure

In the signature generation procedure, we assume that we sign *k*-bit long message, we first split *k* into *l* sub-tree, where $k=2^l$. Then we compute the hash value for the message in the first layer, hash value for the left son's node connecting right son's node, hash value for the message hash value connecting the left son's node and right son's node. For example, compute $N_7=h(m_7)$ for the first layer, $N_{78}=h(h(m_3)||N_7||N_8)$ for the second layer, and $N=H(N_{710}||N_{114})$ for the last layer.

Now we give an example, there are 14 messages. For message m_6, the batch signature is (DS, N_5, N_{58}, N_{14}, N_{914}, N_{114}) for currently batch signature, where the batch residue is (N_5, N_{58}, N_{14}, N_{914}, N_{114}). The batch signature is (DS, N_{112}, N_{710}, N_{714}, h(m_2)) in our scheme, where the batch residue is (N_{112}, N_{710}, N_{714}, h(m_2)).

3) Signature validation procedure

To verify a signature $\sigma = (\sigma_1, \sigma_2,.., \sigma_t)$ on a message *m*, we should imitate the steps which we had in the signature generation procedure. Then we just compare the value of hash function with the corresponding public key. If the values are equal, we think the signature is valid, otherwise invalid.

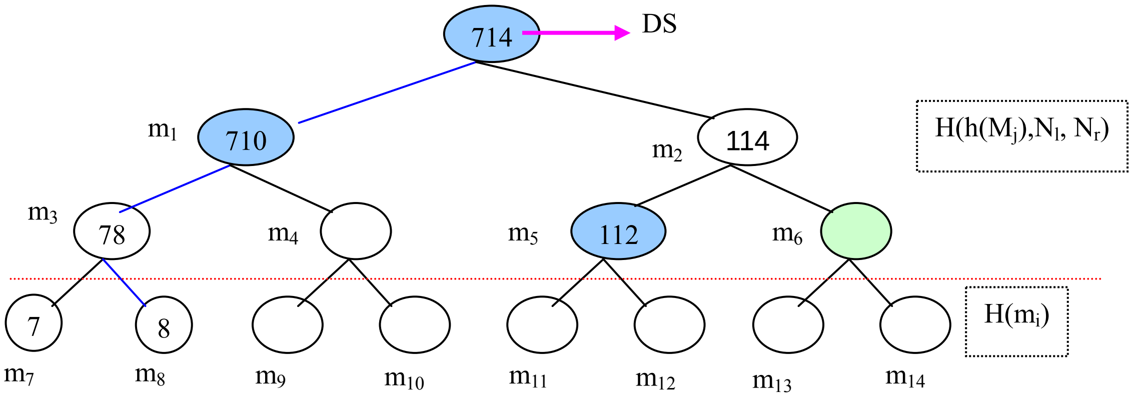

Figure 8. Improved tree-based batch signature

TABLE1
COMPARISON FOR SIZE OF BATCH RESIDUE

Sig.	k=3 |h|=256-bit	k=10 |h|=256-bit	k=20 |h|=256-bit	Size of batch residue
Current batch sig.	6144-bit	320 KB	640MB	$2^k k\|h\|$
Our batch sig.	5642-bit	301KB	608MB	$2^{k-1}(k-2)\|h\|$

Table 1 compares of size of batch residue, we can easily know our signature reduce half size of batch residue, where the parameter is as following: 2^k is the number of message, $|h|$ is the length of hash function.

V. CONCLUSION

Most traditional signature schemes cannot apply in data integrity protection in surveillance video system because of the limited resources of the signer and verifier. This paper proposes a novel method for the data integrity protection in surveillance video system, which can reduce the size of batch residue.

REFERENCE

[1] T. Chen, J. Wang and Y. Zhou. Combined digital signature and digital watermark scheme for image authentication. International Conferences on, Info-tech and Info-net, ICII 2001 - Beijing, Vol. 5, pp. 78-82, 2001.

[2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra. Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol., Vol. 13, No. 7, pp. 560-576, 2003.

[3] Qiu, P. Marziliano, A. T. S. Ho, D. J. He, Q. B. Sun. A hybrid watermarking scheme for H.264/AVC video. In Proc. 17th Int. Conf. Pattern Recogn., U.K., 2004.

[4] J. Zhang, A. T.S. Ho. Efficient video authentication for H.264. IEEE Proc. of the first International Conference on Innovative Computing, Information and Control (ICICIC'06), 2006.

[5] C. V. Nguyen, D.B.H. Tay, G. Deng. A Fast Watermarking System for H.264/AVC Video. IEEE Asia Pacific Conference on Circuits and Syst., APCCAS, pp. 81-84, December 2006.

[6] D. Pröfrock, H.Richter, M. Schlauweg, E. Muller. H.264/AVC video authentication using skipped macroblocks for an erasable watermark. Proc. Of the VCIP, Beijing, China, 2005.

[7] S. Ueda, H. Shigeno, K. I. Okada. NAL Level Stream Authentication for H.264/AVC. IPSJ Transactions on Database, Vol. 48, No. 2, pp. 635-643, 2007.

[8] N. Ramaswamy, K. R. Rao. Video Authentication for H.264/AVC using Digital Signature Standard and Secure Hash Algorithm. NOSSDAV'06, Newport, Rhode Island, USA, May 2006.

[9] K. Ait Saadi, A. Bouridane, A. Guessoum. Combined fragile watermark and digital signature for H.264/AVC video authentication. The 17th European signal Processing Conference (EUSIPCO 2009), Glasgow, Scotland, August, 2009.

[10] A. Pradeep K, Y. Wei Qi, C. Ee-Chien, K. S. Mohan. A hierarchical signature scheme for robust video authentication using secret sharing. in. Proc. IEEE Int. Conf. Multimedia Modelling, 2004.

[11] Christopher J. Pavlovski, Colin Boyd. Efficient batch signature generation using tree structures. Proc. International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC'99), 1999.

[12] William C. Cheng, Cheng Fu Chou, Leana Golubchik. Performance of batch- based digital signatures. Proceedings of IEEE MASCOTS 2002.

[13]R. Johnson, D. Molnar, D. Song, D.Wagner. Homomorphic signature schemes. In B. Preneel, ed., Topics in Cryptology-CT-RSA 2002, Lect. Notes Comp. Sci.., vol. 2771, pp. 244-262. Springer-Verlag, 2002.

[14] S. Haber, W. S. Stornetta. How to Time-Stamp a Digital Document. In Advances in Cryptology-CRYPTO'90, Lect. Notes Comp. Sci., vol. 537, pp. 437-455. Springer- Verlag, 1990.

[15] Miyazaki, K., Susaki, S., Iwamura, M., Matsumoto, T., Sasaki, R., Yoshiura, H. Digital documents sanitizing problem. Tech. Rep. ISEC2003-20, IEICE, 2003.

[16] Steinfeld, R., Bull, L., Zheng, Y. Content extraction signatures. In Information Security and Cryptology-ICISC'01 (Seoul, South Korea, 2002), vol. 2288 of Lecture Notes in Computer Science, Springer-Verlag, pp. 285-304.

[17] Miyazaki, K., Iwamura, M., Matsumoto, T., Sasaki, R., Yoshiura, H., Imai, H. Digitally signed document sanitizing scheme with disclosure condition control. IEICE Trans. Fundamentals E88-A, 1 (2005), 239-247.

[18] Miyazaki, K., Hanaoka, G., Imai, H. Digitally signed document sanitizing scheme from bilinear maps. In The 2005 Symposium on Cryptography and Information Security (SCIS2005) (Maiko Kobe, Japan, 2005), pp. 1471-1476.

[19] Ateniese, G., Chou, D., de Medeiros, B., Tsudik, G. Sanitizable signatures. In ESORICS 2005 (Milan, Italy, 2005), vol. 3679 of Lecture Notes in Computer Science, Springer-Verlag, pp. 159-177.

[20] Merkle, R.C. A certified digital signature. Advances in Cryptology-CRYPTO '89 Proceedings, LNCS 435, pp. 218-238, Springer, 1990.