비디오 감시 장치 무결성 검증을 위한 랜덤 해시 방법

사랄라 기미레, *이범식
조선대학교
srlaghm@chosun.kr, * bslee@chosun.ac.kr (교신저자)

Ramdomg Hash for Integrity Verification Method of Multimedia Data in Surveillance System

Sarala Ghimire, * Bumshik Lee
Chosun University

Abstract

Abstract— The advancement in technology has led to the enormous use of multimedia applications. The video/image recorded by such applications provides critical information that can be used as a visual evidence. However, owing to the adequacy in using different editing tools, it is susceptible to malicious alterations. Thus, the reliability or the integrity of the visual information should be verified before using it as an evidence. In this paper, we propose an integrity verification method for the surveillance system using randomized hashing. The integrity value of the surveillance data is generated using the randomized hashing and elliptic curve cryptography (ECC), which is used later for the validation. The experimental results obtained from the embedded accident data recorder (ADR) system shows that the proposed method is very efficient and provides a high level of security.

1. Introduction

In recent years, the development of digital technology has tremendously increased. Owing to the significance that the digital/physical witness and evidence have, the use of video applications or surveillance systems, such as Closed-Circuit Television (CCTV) systems or accident data recorder (ADR) systems, i.e., the so-called vehicle "black box", has been increased. These systems are typically used in various sensitive areas like monitoring of the activity in the bank, monitoring residential areas, and monitoring the road and highways. The vehicle black box is a device that records the video images in and surrounding of the vehicle in a highway. As the video recorded by the surveillance cameras or the video applications capture critical visual information, which acts as a witness, it plays a crucial role in criminal investigations or dispute examinations [1]. For instance, to investigate the road accident and to determine the cause and victim of the accident, it requires physical evidence such as the status of vehicles, victims, or the real culprit. The traces of such evidence that make easy identification of the cause is obtained by the surveillance cameras mounted publicly on the road or inside the vehicles (ADR). However, the likelihood of alteration of the video content is significantly very high if the offender aims to conceal the decisive information. The openness of the networks [2] that eases the accessibility of the video data is the main reason for the subtle manipulation. The shared and open videos may incur duplication of videos that may infringe the copyright [3] or illegal distribution. In addition, the publicly-available media data can easily be manipulated or tampered with video editing tools utilized with malicious intentions without leaving any visible clues [4]. It is easy to insert or delete certain objects, activity or vehicle information from the current video or can insert frames from other videos. Unlike the surveillance cameras mounted publicly, the probable intruder in ADR is typically the owner or the driver himself. In such a scenario, video data is always exposed to the attacker with a high possibility of video forgery. Thus, the adequacy in performing tampering operation threatens the integrity and authenticity of the video data, particularly if the video is considered as the evidence in criminal or dispute examination in court of law or any other areas like surveillance systems, advertisement and movie industry. It may potentially cause the situation that the wrong person is convicted or punished. Thus, it is important to guarantee that the video content is not forged and is authentic before using it as an evidence.

In this paper, we propose a video integrity verification method for surveillance systems such as ADR and CCTV, based on randomized hashing and the elliptic curve cryptography (ECC) [5] encryption algorithms. Moreover, every one-minute video is randomized with a random salt value before applying a hash algorithm to prevent collision attacks which are then encrypted by using ECC encryption algorithm.

2. Random Hashing and Encryption

The general concept of a hash algorithm is the one-way function i.e. no input can be determined with the given output. Moreover, the arbitrary length input data is mapped to a small fixed-length output, which is designed to provide the integrity of the data. The algorithm is deterministic in nature, that is the same input will always give the same output, while the small change in input sequence must always produce a different output. The design of most of the hash functions follows Markel-Demgard construction [6], where the message is processed iteratively in block-by-block fashion. Due to the block-based iterative design, attacks are typically targeted on the intermediate hash values that generate the colliding message pairs with long message and length extension attacks. The schemes like hash-then-sign or hash-then-encrypt are vulnerable to off-line collision attacks on the underlying hash function. The simple randomization technique, known as randomized hashing (RMX), on the hashing algorithm makes the scheme collision resistant without changing the hash function or the following algorithm [7]. Moreover, RMX provides a hedge against collision attacks on hash functions. Randomization of the video takes the video V and random value k as an input and produces an output v', which is denoted by $RMX\,(v,k)$. $RMX\,(v,k)$ can be defined as the concatenation of the random value k with the result of XOR operation between the value k', generated by concatenating with itself as many times as needed to cover block size string, and every block of the video.

For the protection of key from any attacker, the key is encrypted with asymmetric encryption algorithm. Typically, the key generation process of asymmetric encryption algorithm depends on solving mathematical problems such as factorization of two large integer values or computing discrete logarithms. ECC that employs the point multiplication as a fundamental operation is based on solving the elliptic curve discrete logarithm problem (ECDLP). This problem consists in finding the discrete logarithm q given a point qP. Moreover, it provides a high level of security with smaller key size compared to other cryptographic scheme. The extended form of elliptic curve cryptography known as Elliptic curve integrated encryption scheme (ECIES) is used for the key encryption in the proposed method. ECIES is a hybrid public key encryption that uses the elliptic curve Diffie-Hellman algorithm for key agreement function and symmetric encryption algorithm for encryption of the key.

3. Proposed Method

We propose an integrity verification method for multimedia data based on randomized hashing and key encryption algorithms. It is to be noted that, the videos are recorded in the surveillance system every few minutes interval. For example, CCTV usually records video of every few minutes interval based on the user's setting, whereas the cameras of the ADR system capture the scene of every one-minute interval. In the light of this fact, the video clips recorded every one-minute intervals are defined as video segments in the proposed method. The schematic block diagram of the proposed scheme is shown in Fig. 1, which contains two functions, data integrity function, and integrity validation function. The hash value for each video segments is generated in data integrity function, whereas the integrity validation function verifies the integrity of the data.

The data integrity function generates hash value of each video segment using randomized hashing and encrypts the hash value along with the key using ECC encryption algorithm. Initially, the random keys are generated from random function. The video v is then partitioned into v_1 to v_l of b-bit blocks, where l is the number of video blocks. The partitioned blocks are randomized with random unique values r.

$$v_i = v_i \oplus r_i, \quad i = 1 \text{ to } l \qquad (1)$$

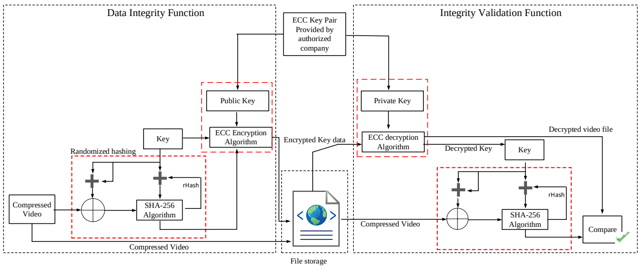

Fig 1. Block diagram of the proposed method.

The output from this randomization process is given as input to the SHA-256 function that gives *rHash* value, which is concatenated with secret key k and hashed with SHA-256 that gives *fHash*. Finally, *fHash* and the key k are encrypted with ECC encryption algorithm and stored in the storage as shown in fig.1. Thus, since the input message is randomized with the unique random value before applying hash function, the attack by appending new video segment to the original video is not possible. Similarly, it is hard for the adversary to find the video v' and random value r' such that, given a uniform random value r and the video v, they can find $(v,r) \neq (v',r')$ but $H\,(v,r) = (v',r')$. Furthermore, the second stage of hashing along with the key confirms the authenticity of the data as well as makes the internal state non-constructible. The final encryption of the hash value and the key provides further security to the video segment.

The video segment submitted for the validation is validated in the integrity validation function. Initially, the encrypted hash and the key stored in the storage are extracted and decrypted with ECC private key. The integrity of the video is verified by comparing the decrypted hash value with the currently generated hash of the corresponding video segment. The current video segment is randomized with the random values generated from random function by following the same process as in data integrity function. The randomized output is then input to the SHA-256. The output from the first hash (*rHash*) is concatenated with the decrypted key *k* and hashed again with second hash function SHA-256 that gives *fHash*. At the final stage, the hash value *fHash* is compared with the decrypted hash value *fHash*'. The video segment is considered untampered if the two hash values are matched else the integrity is invalid.

4. Experimental Results

To evaluate the proposed method, we used eight publicly available video clips with resolution 1280×720 pixels and frame rate 30 frames per second (fps). The test videos are forged by adding forgery with the commercial video editor, the AVS video editor. Tampering is computed by randomly inserting, copying and pasting, and deleting frames in the video segments. The eight test videos with several frames and tampering type are listed in Table I, which also shows that the proposed method can detect any forgery type.

First, the raw video segments were encoded using H264/AVC compression technique. The compressed stream is randomized, hashed and encrypted along with the key. During validation, the stored video bitstream is hashed and encrypted in the same way, which is then compared with the decrypted hash value. Thus, the experimental analysis is performed on an Ambarella board of a 792-MHz ARM® Cortex™-A9 CPU, DDR3 / DDR3L with up to 600 MHz memory, which is an embedded board for the ADR made by Ambarella Inc.

The analysis based on the running time for both integrity and validation function on various length videos with and without randomization is shown in Fig. 2 and Fig. 3. As shown in Fig. 2 and Fig. 3, the proposed method with randomization drains about 100 to 500 milliseconds more time than the conventional hash algorithm, which is significantly very less. The extra process of randomization is the reason for this nominal increase that introduces the computational overhead of less than 2.5%. In light of these results, it shows that the proposed method can be applied in real-time scenarios.

TABLE I
VIDEO FILES USED IN THE DETECTION TEST

Video	Original Frames	Tampering Type	Tampered Frames	Time length (sec)	Forgery Detection
1	450	Copy-paste	450	15	Yes
2	900	Copy-paste	900	30	Yes
3	1797	Insert	1800	60	Yes
4	3593	Insert	3600	120	Yes
5	4867	Delete	4800	160	Yes
6	8720	Copy-paste	8720	290	Yes
7	10910	Copy-paste	10910	360	Yes
8	15120	Copy-paste	15120	510	Yes

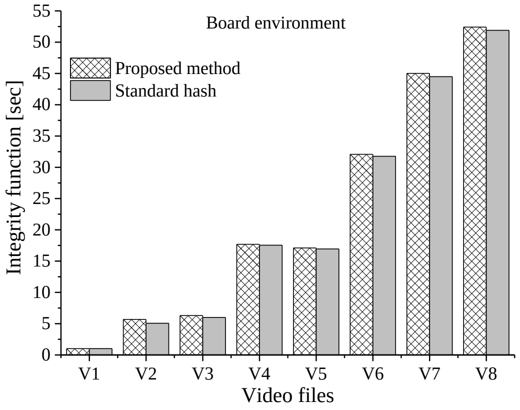

Fig 2. Time comparison of integrity function for various sized videos in Ambarella board.

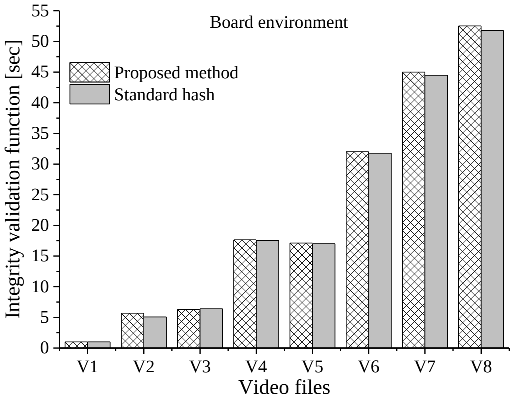

Fig 3. Time comparison of integrity validation function for various sized videos in Ambarella board.

5. Conclusion

In this paper, we proposed an integrity verification method for surveillance system. The randomization on the video data before hash function prevents a hash collision attack. The additional encryption process for the key and the randomized hash value provides further security to the video data. The comprehensive experimental results show that the scheme is more efficient in terms of execution speed and applicable to real-world consequences.

Acknowledgement

This research was supported by the MIST (Ministry of Science & ICT, Korea, under the National Program for Excellence in SW) supervised by the IITP (Institute for Information & communications Technology Promotion)" (2017-0-00137)

References

[1]C. Lee, J. Lee, Y. Pyo, and H. Lee, "Broken Integrity Detection of Video Files in Video Event Data Recorders,"

KSII Trans. Internet Inf. Syst., vol. 10, no. 8, pp. 3943-3957, 2016.

[2] X. Nie, Y. Yin, J. Sun, J. Liu, and C. Cui, "Comprehensive Feature-Based Robust Video Fingerprinting Using Tensor Model," IEEE Trans. Multimed., vol. 19, no. 4, pp. 785–796, Apr. 2017.

[3] C. Chou, H. Chen, and S. Lee, "Pattern-Based Near-Duplicate Video Retrieval and Localization on Web-Scale Videos," IEEE Trans. Multimed., vol. 17, no. 3, pp. 382-395, Mar. 2015.

[4] L. Yu et al., "Exposing frame deletion by detecting abrupt changes in video streams," Neurocomputing, vol. 205, pp. 84–91, Sep. 2016.

[5] D. Hankerson, A. Menezes, and S. V. Springer, "Guide to Elliptic Curve Cryptography."

[6] I. B. Damgaard, "A Design Principle for Hash Functions," in Advances in Cryptology — CRYPTO' 89 Proceedings, New York, NY, 1990, pp. 416–427.

[7] S. Halevi, H. Krawczyk, D. Boneh, and M. Mcintosh, " Implementing the Halevi-Krawczyk Randomized Hashing Scheme," pp. 1–15.