2021

An Anti-Recompression Video Watermarking Algorithm in Bitstream Domain

Jing Sun  
*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*

Xiaoping Jiang  
*College of Electrical and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China*

Jin Liu  
*Shenzhen Beidou Applied Technology Research Institute, Shenzhen 518055, China*

Fan Zhang  
*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; Shenzhen Beidou Applied Technology Research Institute, Shenzhen 518055, China*

Congying Li  
*College of Electrical and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China.*

Follow this and additional works at: [https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology](https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology)

Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

**Recommended Citation**

Jing Sun, Xiaoping Jiang, Jin Liu et al. An Anti-Recompression Video Watermarking Algorithm in Bitstream Domain. Tsinghua Science and Technology 2021, 26(2): 154-162.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Tsinghua Science and Technology by an authorized editor of Tsinghua University Press: Journals Publishing.
An Anti-Recompression Video Watermarking Algorithm in Bitstream Domain

Jing Sun*, Xiaoping Jiang, Jin Liu, Fan Zhang, and Congying Li

Abstract: Following the popularity of digital video application, video copying and dissemination have become very easy; however, this makes video hacking and piracy a potential threat in video communication. Video watermarking technology can solve the problem of copyright protection, and thus, it has been extensively researched. The robustness of the video watermarking algorithm in the bitstream domain is poor, especially the anti-recompression ability, since the watermarked video may be compressed again before transmitting. Considering this, this paper proposes a video watermarking algorithm in the bitstream domain based on moving object detection. To increase the robustness of the watermarking scheme, the moving macroblocks that belong to the moving object in each P frame can be identified via the moving object detection algorithm. Then, watermark embedding in the moving macroblocks is performed using codeword substitution to ensure the consistency of the bitstream. Moving object detection and watermark embedding are independent and are both carried out in the bitstream domain by partially decoding the bitstream; this avoids the complete decoding and reconstruction of the video, making the method to be highly efficient. The simulation results confirm that the proposed method is robust against recompression and has little impact on the video visual quality and no influence on the bit rate.

Key words: video watermarking; recompression; moving object detection

1 Introduction

With the developments of computer technology and cloud computing technology, the generation, storage, and dissemination of digital video products have become more convenient. Meanwhile, with the extensive popularity of the Internet, digital video application has become widespread in people’s daily life. However, this is followed by illegal acts such as piracy, infringement, and tampering, which greatly damage the property rights of the digital work owner[1]. Video watermarking can truly solve the problem of copyright protection for digital products; consequently, video watermarking technology has become a research hotspot[2].

In recent years, various video watermarking algorithms have been presented[3–9]. There are three main categories of watermarking in video codec: video watermarking in the spatial domain, video watermarking during the encoding process, and video watermarking in the compressed domain[9]. The first two need coding reconstruction, and any modification during the watermark embedding might likely affect the further encoding of information, while the last embeds a watermark in the compressed domain (syntax elements...
before entropy encoding) or bitstream domain (bit substitutions of the final stream). Video watermarking in the compressed domain requires either only the entropy encoding stage or no stages from the compression. Therefore, it is very runtime-efficient compared to the first two. In this study, we focus on video watermarking in the bitstream domain.

Watermark embedding may be conducted directly on the bitstream, i.e., by substitution of certain bits. Kim et al.\textsuperscript{[10]} proposed an algorithm that embeds the watermark in the sign bit of the TrailingOnes in Context-Adaptive Variable-Length Coding (CA VLC). Zou and Bloom\textsuperscript{[11]} described a method for applying a watermark directly to an entropy-coded H.264/AVC stream; the watermark is embedded in the Intra-Prediction mode through mb type, by changing the mb type in the stream to another value that results in a compliant bitstream. Xu et al.\textsuperscript{[12]} extended the embedded position by using a codeword-substituting method through CA VLC codeword mapping and Exp-Golomb codeword mapping. Based on exploiting the redundancy existing in CA VLC codeword, Xu et al.\textsuperscript{[13]} proposed an improved version that is indeed capable of providing a larger embedding capacity than the previous methods. The methods in Ref. \textsuperscript{[13]} show great advantages as they are very efficient and maintain the format compliance; however, they have poor robustness to recompression.

Based on this, we propose an anti-recompression video watermarking algorithm. To provide a more efficient watermarking, the watermarking algorithm is conducted directly on the bitstream. To increase the robustness of the watermarking scheme, only a set of macroblocks that belong to moving objects is chosen for inserting the watermark. The object-based watermarking might be robust because it is difficult to damage the watermark without affecting the visual quality of the foreground video objects which are very susceptible to the human visual system, and any changes in the objects are easily perceptible\textsuperscript{[2]}. Moving objects blocks are detected from the video bitstream using a moving object detection algorithm\textsuperscript{[14, 15]}, which is independent of watermark embedding. To keep the bitstream compliant, the embedding process adopts the codeword substitution with the same length, avoiding an increase in bit rate. Meanwhile, the pair of the replacement codewords have little difference to maintain the embedding imperceptibility.

The remainder of the paper is organized as follows. In Section 2, we describe the proposed scheme, which includes three parts: moving object detection, watermark embedding, and watermark extraction. The experimental results and discussions are presented in Section 3. Finally, conclusion is drawn in Section 4.

## 2 Proposed Scheme

In this section, an anti-compression watermarking algorithm based on moving object detection is presented, which includes three parts: moving object detection, watermark embedding, and watermark extracting.

### 2.1 Moving object detection

Moving object detection is performed on the compressed bitstream file of the video, and the purpose is to select the moving objects for each frame to obtain the corresponding motion blocks which belong to objects.

Specifically, we partially decode the compressed bitstream to obtain the Motion Vectors (MVs) of each macroblock, and then a certain operation is performed using all the MVs of each macroblock to obtain a value that is compared with another value named threshold \( T \), and then, the compared results are used to determine whether the macroblock belongs to the moving objects. Taking H.264 as an example, we detail the steps in the following.

**Step 1:** Let the size of the current macroblock be \( W \times H \), where \( W \) and \( H \) are respectively the width and height of the macroblock, which is determined by the encoding rule to be 16\times16 in H.264/AVC. The current macroblock can be divided into several sub-macroblocks, and each sub-macroblock can get an MV. The minimum size of a sub-macroblock in H.264/AVC is 4\times4. If the current macroblock is divided into \( N \) sub-macroblocks, there will be \( N \) MVs. The \( N \) MVs of each macroblock will be obtained by partially decoding the compressed bits stream.

**Step 2:** For \( N \) MVs in each macroblock, sum the norms of all MVs, and divided by the area of the macroblock, and then divide by 2 since each MV is a two-dimensional array. The calculated value is called \( \text{MV}_{\text{MB}} \) which is described in Eqs. (1) and (2).

\[
\text{MV}_{\text{MB}} = \frac{\sum_{i=1}^{N} ||\text{MV}_i||}{2 \times W \times H}, \quad i = 1, 2, \ldots, N \quad (1)
\]

\[
||\text{MV}_i|| = \sqrt{\text{MV}_x^2 + \text{MV}_y^2} \quad (2)
\]

where \( \text{MV}_x \) and \( \text{MV}_y \) are the horizontal and vertical components of a motion vector, respectively.

**Step 3:** Compare the obtained \( \text{MV}_{\text{MB}} \) with the
threshold $T$. If $\text{MV}_{\text{MB}}$ is greater than $T$, the current macroblock belongs to the moving objects, otherwise, it belongs to non-moving objects. Here, $T$ is related to the motion intensity of the video sequence. The larger the $T$ value, the better the invisibility of the watermark, but the corresponding embedding capacity will be smaller. The smaller the $T$ value, the larger the capacity of embedded watermark.

**Step 4:** Perform Steps 2 and 3 for each macroblock until all the macroblocks belonging to the moving objects of each frame have been selected.

### 2.2 Watermark embedding process

Watermark embedding is performed in the bitstream domain. In this paper, CA VLC is used as the entropy coding, and the watermark will be embedded in the codewords of Level obtained after entropy encoding in the macroblocks that belong to the moving objects of the P-frames of the video.

In H.264/AVC baseline profile, each CA VLC codeword is composed of five parts as follows: 
\{
\text{Coeff}_{\text{token}}, \text{Sign}_0\text{f}_\text{TrailingOnes}, \text{Level}, \text{Total}_0\text{zeros}, \text{Run}_{\text{before}}\}.
The specific function of each part is described in Ref. [16]. In this paper, the codeword of Level within P-frames is used as the carrier to embed watermark. If the watermark is embedded in I-frame, the image quality as well as the subsequent P-frame, will be affected, so we choose P-frame to embed the watermark. To keep the bitstream compliant, the codeword of the Coeff_token, Total_zeros, and Run_before cannot be modified during the watermark embedding process. The codeword of Sign_of_TrailingOnes can only carry 1 bit watermark, and changing the codeword of the 1 bit will greatly impact on video quality. In contrast, the codeword of Level is more suitable for watermark embedding since the codeword substitution in the codeword of Level can not only keep the bitstream compliant but also has little influence on the video quality, and the capacity of watermarks is also proper.

The codeword of Level consists of Level_pre and Level_suffix. SuffixLength is used to indicate the length of Level_suffix. By analyzing the relationship of Levels and their corresponding codewords, when the Suffixlength is equal to 2 or 3, we implement a watermark embedding operation in the codeword of Level by codeword substitution. The reasons we choose Suffixlength as 2 or 3 are as follows: When Suffixlength is 0, we cannot find a substituted codeword of Level that has the same size as the original one. When Suffixlength is 1, although the codeword of Level with the same length can be found, choosing this Suffixlength is equivalent to reversing the value of Level, which has a great effect on the video quality. When Suffixlength is equal to 2 or 3, a pair of the codeword of Level with the same length can be used to form codeword substitution map to remain syntax compliance, and the effect on the Level value is only +1 or −1, which has little effect on the quality of the video.

The detailed steps for embedding the watermark are as follows.

**Step 1:** Parse the bitstream and obtain the codewords of Level.

**Step 2:** Judge each macroblock whether it satisfies the condition, and the condition is that the current macroblock within P-frames belongs to moving objects and the Suffixlength is equal to 2 or 3.

**Step 3:** For the current macroblock that meets the embedding condition, the embedding watermark operation is to be performed. According to the value of the watermark that needs to be embedded, look up the codeword substitution map of Level to perform codeword substitution. The codeword substitution map and the specific operation of codeword substitution are explained as follows.

Figure 1 lists some as an example of the codeword substitution map. In it, T0 indicates that the bit of the watermark that needs to be embedded is “0”, and T1 indicates that the bit of the watermark that needs to be embedded is “1”, and it is also useful to reverse them. Based on the above definition, the codeword substitution can be detailed as follows: If the bit of the watermark that needs to be embedded is “0” and the codeword of Level as a carrier belongs to the T1, or if the bit of the watermark that needs to be embedded is “1” and the codeword of Level as a carrier belongs to T0, the codeword of Level will be replaced by the corresponding value from the codewords substitution map; otherwise, the codeword of Level remains unchanged. For example, if the bit of the watermark that needs to be embedded is

| SuffixLength | Level | Codeword | T0  | Codeword | T1  |
|-------------|-------|----------|-----|----------|-----|
| 2           | 1     | 100      | 110 | 2        | 2   |
| 2           | -1    | 101      | 111 | -2       | 2   |
| 2           | 3     | 0100     | 010 | 4        | 2   |
| 2           | -3    | 0101     | 011 | 4        | 2   |
| 3           | 1     | 1000     | 1010| 2        | 3   |
| 3           | -1    | 1001     | 1011| -2       | 3   |
| 3           | 5     | 01000    | 01010| 6       | 3   |
| 3           | -5    | 01001    | 01011| -6      | 3   |

Fig. 1 Codewords substitution map.
“0” and the codeword of Level as a carrier is “110” which belongs to T1, the codeword “110” should be replaced with “100”.

**Step 4:** Repeat until all frames have been processed.

An example of watermark embedding based on codewords map is shown in Fig. 2a. Here, suppose the bits of the watermark that needs to be embedded is “1010”, the CAVLC codewords of Level parsing from H.264/AVC bitstream are “100 0100 0101 1010 01001 01011”. According to the watermark embedding method, the codewords after watermark embedding are “100 0110 0101 1010 01001 01001”.

### 2.3 Watermark extraction process

In our scheme, the watermark extraction process is fast and simple, and it is also a blind process as it does not require using the origin video. The extraction process is given as follows.

**Step 1:** Parse the bitstream and identify the codewords of Level.

**Step 2:** Judge each macroblock whether it satisfies the condition shown in watermark embedding process.

**Step 3:** For each macroblock that meets the condition, the watermark extraction operation is to be performed. The operation depends on codeword substitution map of the watermark embedding process. If the codeword of Level belongs to T0, the extracted watermark bit is “0”. If the codeword of Level belongs to T1, the extracted watermark bit is “1”.

**Step 4:** Repeat until all frames have been processed.

An example of watermark extraction in the bitstream domain is shown in Fig. 2b. According to the watermark extraction method, the extracted watermark information is “1010”.

### 3 Experimental Results and Discussions

#### Intermediate

The experiments are presented in two parts: one is the experimental results of the moving object detection, the other is the experimental results of the watermark. The method described in this paper is simulated in the environment of H.264/AVC reference software JM8.6 and Matlab R2014a, and the following Common Intermediate Format (CIF) is selected: (352×288, YUV 4:2:0), standard video sequences: soccer, stefan, hall, foreman, news, tempete, and tennis. The baseline profile is adopted, 50 frames are coded in total, the Group Of Picture (GOP) structure is IPPP, and Quantizer Parameter (QP) is set to 24, as shown in Table 1.

#### 3.1 Motion block detection

Figure 3 is the effect diagram obtained by moving object detection algorithm. The sign of the box denotes that the current macroblock belongs to the moving objects. After identifying these macroblocks belonging to moving objects, we will embed the bits of the watermark through the watermark embedding algorithm.

#### 3.2 Imperceptibility

In this section, we will present the experimental results of the watermark. Imperceptibility, also called invisibility, requires that the watermark embedded in the video cannot significantly affect the video visual quality. The Mean Peak Signal-to-Noise Ratio (MPSNR) and Mean Structural SIMilarity (MSSIM) are used to quantitatively evaluate the imperceptibility of the video watermarking algorithms. If the number of the watermarked frames is \( K \), and the size of the video frame is \( M \times N \), the MPSNR definition is given as follows:

\[
\text{MPSNR} = \frac{1}{K} \sum_{k=1}^{K} \text{PSNR}_k, \quad k = 1, 2, \ldots, K
\]

\[
\text{PSNR}_k = 10 \log_{10} \frac{255^2}{\text{MSE}_k}
\]

![Fig. 2](image.png) Example of watermark embedding and extraction.
where $I_k$ is the $k$-th frame of the original video and $I_{kw}$ is the $k$-th watermarked frame. The definition of MSSIM is shown as below:

$$\text{MSSIM} = \frac{1}{K} \sum_{k=1}^{K} \text{SSIM}_k, k = 1, 2, \ldots, K$$

$$\text{SSIM}_k = \frac{(2u_1u_2 + C_1)(2\sigma_{12} + C_2)}{(u_1^2 + u_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)}$$

where $u_1$ and $u_2$ are the mean values of the original frame and the watermarked frame, respectively; $\sigma_1$ and $\sigma_2$ are the variances of the original frame and the watermarked frame, respectively; $\sigma_{12}$ represents the covariance of the original frame and the watermarked frame; and $C_1$ and $C_2$ are two constants.

We compared the watermarked video with the original one to subjectively evaluate the imperceptible performance. Figure 4 presents the frames of videos before and after watermark embedding; there are little differences between the watermarked frame and the original one.

Table 2 shows the MPSNR and MSSIM comparison between the original videos and the watermarked ones.

---

Fig. 3 Moving object detection.

Fig. 4 Frames of videos before and after watermark embedding.
by our proposed scheme. From Table 2, it can be found that the proposed scheme has little influence on the videos, and the algorithm can meet the human perceptual requirement. The main reason is that the codeword substitution we adopted has an effect of only +1 or −1, which has little influence on the visual degradation of the video. In addition, watermarks are embedded only into a small portion of the codewords of the moving objects macroblocks, indicating a great control of the embedding capacity, thus ensuring the video quality.

### 3.3 Robustness against recompression

Robustness against recompression of the algorithm can be evaluated by the Bit Error Rate (BER). The definition of BER is shown in Eq. (8),

$$ BER = \frac{\sum_{i=1}^{S_n} \omega'_i \oplus \omega_i}{S_n} $$

where $S_n$ is the total number of bits of the embedded watermark, $\omega'_i$ is the extracted watermark sequence, and $\omega_i$ is the original watermark sequence. The smaller the BER, the better the robustness.

We verified anti-recompression performance of our watermarking scheme by recompression. Figure 5 shows the BER between the original watermark and the extracted one after one-time recompression of the watermarked video by H.264 codes. The results of Fig. 5 show the BER value of the extracted watermark from the method in Ref. [12] increased dramatically after recompression, in contrast, the BER from the proposed scheme was very small after recompression.

We also present more experiments involving multiple recompressions of the watermarked video. Figure 6 shows the experimental results of the video sequences, such as soccer, stefan, hall, foreman, news, and tempete. In Fig. 6, for one-, two-, and three-time recompressions of the watermarked video from the proposed scheme, the BER value between the original watermark and the extracted one was small. In contrast, the BER value grew large after the one-, two-, and three-time recompressions of the watermarked video from the method in Ref. [12]. This confirms that the proposed scheme has a robust anti-recompression performance.

### 3.4 Embedding capacity

The watermark capacity illustrates the number of watermark bits embedded in unit time or in a single video. Figure 7 gives the embedding capacity in P-frames of the video sequences by our scheme. The horizontal coordinate represents frame number, and the vertical coordinate represents the embedded bits of each frame. It is common that the watermark capacity, imperceptibility, and robustness are mutually restricted. According to different practical requirements, a tradeoff occurs between the capacity and the visual quality if you want to get a higher embedding capacity.

### 3.5 Bit rate variation

Bit rate is used to measure the increase of the video bit rate, which is often measured in watermarking algorithms based on compressed videos. In the paper, the proposed scheme is performed by replacing a suitable codeword with another codeword of the same length, so the bit rate of the watermarked video remains unchanged, and the watermark embedding process does not affect the compression efficiency of the encoders.

### 4 Conclusion

In this paper, an anti-recompressed video watermarking algorithm based on moving object detection in the bitstream domain is proposed. We use the H.264/AVC compression standards to simulate our method. To increase the robustness of the watermarking scheme against anti-compression, we first identify the macroblocks belonging to moving objects, and then, the watermark is embedded in the codewords obtained after entropy encoding in these identified macroblocks. To keep the bitstream compliant, as well as satisfy the perception requirement of the watermarked video, the embedding process adopts the codeword substitution

| Original video | MPSNR (%) | MSSIM (%) |
|----------------|-----------|-----------|
| Hall           | 41.70     | 98.39     |
| Stefan rate    | 42.79     | 99.23     |
| Foreman        | 39.89     | 96.87     |
| Tempete        | 42.96     | 98.97     |
| News           | 43.15     | 98.60     |
| Soccer         | 40.35     | 96.29     |
method using a pair of codeworks with the same length and little difference. We measure the transparency and robustness of our proposed method by simulation. We also verify the anti-recompression performance of the propose method. The moving object detection and watermark embedding in the proposed method are performed in the video bitstream domain. We only partly decode the bitstream to obtain the needed codewords, which improves the efficiency of watermark embedding, and makes it suitable for applications in video-based real-time monitoring.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 61702563).

References

[1] W. Lu, A. L. Varna, and M. Wu, Secure video processing: Problems and challenges, in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 2011, pp. 5856–5859.

[2] M. Asikuzzaman and M. R. Pickering, An overview of digital video watermarking, IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, pp. 2131–2153, 2018.

[3] X. Yu, C. Wang, and X. Zhou, A survey on robust video watermarking algorithms for copyright protection, Applied Sciences, vol. 8, no. 10, pp. 1891–1916, 2018, doi:10.3390/app8101891.

[4] A. Tiwari, S. K. Ojha, and D. Pantiukhin, An effective approach for secure video watermarking based on H.264 coding standard, in Proc. of IEEE International Conference on Computational Intelligence and Communication Technology, Kolkata, India, 2017, pp. 1–5.

[5] N. Zarmehi and M. J. Barikbin, Robust video watermarking against H.264 and H.265 compression attacks, Frontiers in Signal Processing, vol. 1, 2017.

[6] X. Li, X. Wang, W. Yang, and X. Wang, A robust video watermarking scheme to scalable recompression and transcoding, in Proc. of IEEE International Conference on Electronics Information and Emergency Communication, Beijing, China, 2016, pp. 257–260.

[7] G. Kaur, S. S. Kasana, and M. K. Sharma, An efficient watermarking scheme for enhanced high efficiency video coding/h.265, Multimedia Tools and Applications, vol. 78, pp. 12537–12559, 2018.

[8] C. Y. Wang, R. Y. Shan, and X. Zhao, Anti-HEVC recompression video watermarking algorithm based on the all phase biorthogonal transform and SVD, IETE Technical Review, vol. 35, no. sup1, pp. 1–17, 2018.

[9] W. Chen, Z. Shahid, T. Stütz, and F. Autrusseau, Robust
Jing Sun et al. An Anti-Recompression Video Watermarking Algorithm in Bitstream Domain

Fig. 7 Embedding capacity in P-frames.

Drift-free bit-rate preserving H.264 watermarking, Multimedia Systems, vol. 20, no. 2, pp. 179–193, 2014.

[10] S. M. Kim, B. K. Sang, Y. Hong, and C. S. Won, Data hiding on H.264/AVC compressed video, in Proc. of International Conference on Image Analysis and Recognition, Montreal, Canada, 2007, pp. 698–707.

[11] D. Zou and J. A. Bloom, H.264/AVC stream replacement technique for video watermarking, in Proc. of IEEE International Conference on Acoustics, Las Vegas, NV, USA, 2008, pp. 1749–1752.

[12] D. Xu, R. Wang, and Y. Q. Shi, Data hiding in encrypted H.264/AVC video streams by codeword substitution, IEEE Transactions on Information Forensics & Security, vol. 9, no. 4, pp. 596–606, 2014.

[13] D. Xu, R. Wang, and Y. Q. Shi, An improved scheme for data hiding in encrypted H.264/AVC videos, Journal of Visual Communication and Image Representation, vol. 36, pp. 229–242, 2016.

[14] R. V. Babu, M. Tom, and P. Wadekar, A survey on compressed domain video analysis techniques, Multimedia Tools & Applications, vol. 75, pp. 1043–1078, 2016.

[15] J. D. Praeter, J. V. D. Vyver, N. V. Kets, and G. V. Wallendael, Moving object detection in the HEVC compressed domain for ultra-high-resolution interactive video, in Proc. of IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 2017, pp. 135–136.

[16] Sofokleous and Anastasis, Review: H.264 and MPEG-4 video compression: video coding for next-generation multimedia, The Computer Journal, vol. 48, no. 5, pp. 563–5638, 2005.

Jing Sun received the PhD degree from Wuhan University, China, in 2013. She is currently a postdoctoral fellow at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Her research interests include multimedia processing, multimedia security, and pattern recognition.

Jin Liu received the PhD degree from Wuhan University, China in 2011. He is currently a senior engineer at Shenzhen Beidou Applied Technology Research Institute, Shenzhen, China. His research interests include multimedia processing, cloud computing, and big data processing.
Xiaoping Jiang received the PhD degree from Huazhong University of Science and Technology, China in 2007. He is currently an associate professor at the College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan, China. His research interests include video analysis, pattern recognition, and wireless communication.

Fan Zhang received the PhD degree from Huazhong University of Science and Technology, China in 2007. He is currently a professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is also with Shenzhen Beidou Applied Technology Research Institute, Shenzhen, China. His research interests include cloud security, big data processing, and data mining.

Congying Li is a master student at the College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan, China in 2019. His research interests include signal process and multimedia security.