International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 5
ISSN: 2454-4248
76 – 80

An Improved Watermarking Algorithm Robust to Temporal Desynchronization Attacks

Yiqun Hui
Department of Computer Science and Technology
Tsinghua University
Beijing, China
huiyq15@mails.tsinghua.edu.cn

Daoshun Wang
Department of Computer Science and Technology
Tsinghua University
Beijing, China
daoshun@mail.tsinghua.edu.cn

Xianghui Zhao
China Information Technology Security Evaluation Center
Beijing, China
zhaoxianghui@mail.itsec.gov.cn

Shundong Li
School of Computer Science
Shanxi Normal University
Shanxi, China
shundong@snnu.edu.cn

***Abstract*—**The watermarking scheme based on finite state machine can achieve temporal desynchronization without additional synchronization signals or exhaustive search. However, the resistance to the frame dropping attack and the decimation attack is highly dependent on the repetition redundancy unit. When the video is truncated, permuted and reassembled, the extracted watermark has a low bit correct rate (BCR). In this work, we analyze and improve the correlation-based extracting method for spread-spectrum watermarking and propose a non-synchronization extracting strategy based on statistical inference. The proposed method is resistant to most of the temporal synchronization attacks.

Keywords*-video watermarking; temporal desynchronization attacks; spread-spectrum watermarking; statistical inference*

*****

I. INTRODUCTION

The temporal synchronization is the process to find the location of the watermark signal, which is very important for blind extraction schemes.

When the video content is processed or transmitted, the frame rate may be converted to improve the compatibility of the terminal clients or the efficiency of the storage and the bandwidth. The frame rate conversion is usually achieved in two ways:

- Frame duplication is performed to increase the frame rate and frame dropping is performed to decrease the frame rate. These operations are called the duplicating attack and the dropping attack.
- Frame interpolation is performed to increase the frame rate and frame averaging is performed to decrease the frame rate. These operations are relatively computationally intensive, but applicable when the quality is concerned, which are called the averaging attack and the inserting attack.

A practical watermarking scheme should resist these temporal desynchronization attacks. Several different strategies have been utilized to achieve temporal synchronization.

The watermarking algorithms achieving synchronization by exhaustive search must explicitly search the spaces of coordinate transformations to locate the watermark, which is extremely computationally expensive. However, there are schemes that limit the searching space[1].

Some other works prefix a synchronization signal to the watermark signal during the embedding process and detect the synchronization signal before extraction [2],[3]. This method is intuitive and efficient, but the frame dropping and frame inserting may still distort the extracted watermark.

To achieve better synchronization, [4] develops models for watermark embedding and detection to examine the temporal synchronization for blind video watermark detection. They propose an embedder modeling the construction of the watermark by using a state machine key generator and a feature extractor. The generated watermark is proved easy to extract with a state predictor and a queue even through desynchronization attacks.

This synchronization framework is then combined with the spread-spectrum watermarking [5]. In this scheme, the spread-spectrum code, by nature a key for embedding and extracting, is generated with the state machine key generator and the feature extractor. This scheme can resist most of the temporal desynchronization attacks and spatial noise attacks. However, it's figured out that the watermark detector must determine the key when a frame is examined [6], which does not hold in the spread-spectrum watermarking. The correlation-based watermark detector for spread-spectrum watermarking can speculate whether the given frame is watermarked with the specific key, but the result may be incorrect. There is still room left for improvement.

In this paper, we develop the probability model of the spread-spectrum watermarking and improve the correlation-based extracting method to increase the accuracy. In addition, we also propose a non-synchronization extraction strategy based on statistical inference to resist the temporal desynchronization attacks more thoroughly. The experiments show the effectiveness and performance of our methods.

II. SPREAD-SPECTRUM WATERMARK

The idea of spread-spectrum watermarking comes from the spread-spectrum communication in the telecommunications field [7]. In general, the bandwidth is limited, so most

 ISSN: 2454-4248
76 – 80

communication systems use as less bandwidth as they need; while the principle of the spread spectrum is opposite, it always occupies several times the minimum bandwidth. It is widely used to establish secure channels and increase resistance to natural interference.

In the field of watermarking, the spread-spectrum watermarking refers to the watermarking scheme which spreads meaningful watermark signals into noise-like spread-spectrum signals with wider bandwidth before embedding. Therefore, it's difficult for an attacker without the spread-spectrum code to interpret the watermark signal even if it has been extracted. Besides, since the watermark signal is spread over a wider spectrum, the noise signal distorts the watermark signal less.

In summary, the spread-spectrum watermarking technology improves the robustness and security at the expense of capacity.

A. *Embedding Process*

Verious forms of spread-spectrum watermarking have been proposed [7], while the additive spread-spectrum watermarking is most widely used by the following works [8]-[10], which is expressed as:

$$\tau: F \to V. \tag{1}$$

$$V' = V + \alpha W. \tag{2}$$

$$\tau^{-1}: V' \to F'. \tag{3}$$

Specifically, a video frame F is converted into corresponding transform domain coefficients V by a transformation τ, and then the spread-spectrum code W is added to the transform domain coefficients with a watermark strengthα. W is a pseudo-random sequence consisting of 1 and -1 with the same length as V. It also satisfies that the mean$\bar{W}$is zero.

There are a variety of transformations have been chosen in spread-spectrum watermarking, including Discrete Fourier Transformation (DFT), Discrete Wavelet Transformation (DWT) and Discrete Cosine Transformation (DCT). With different transformation, the spread-spectrum watermarking can achieve different visual effects and persistdifferent spatial attacks. The transformation is not discussed in this work to concentrate on the temporal attacks.

B. *Original Extracting Method*

The correlation coefficient between the transform domain coefficients V'' of the attacked frame and the spread-spectrum code W is commonly utilized to extract the watermark:

$$r = Corr(V'', W). \tag{4}$$

Given a threshold T_r, if r satisfies:

$$r > T_r. \tag{5}$$

Then the frame is inferred watermarked.

To distinguish watermark bit 0 and 1, algorithms always generate two different spread-spectrum codes to embed with during embedding and detect the watermark with the two different spread-spectrum codes when extracting. The two spread-spectrum codes are either uncorrelated or with a correlation coefficient of -1.

As the attackers have no prior knowledge of W, the attacks applied to the watermark are regarded as a random noise A uncorrelated with W:

$$V'' = V' + A = V + \alpha W + A = (V + A) + \alpha W. \tag{6}$$

Let the $\cdot$ denotes the dot product of two arrays, the $||$ denotes the 2-norm of the array, the $\vee$ denotes the angle between two arrays, the $\bar{X}$ denotes the average of X, and$\check{X}$denotes the centered array of X(or$\check{X} = X - \bar{X}$), then the correlation coefficient is calculated as:

$$Corr(A, B) = \frac{\check{A} \cdot \check{B}}{|\check{A}||\check{B}|} = cos\, \check{A} \vee \check{B} \tag{7}$$

So, the correction coefficient between two arrays is equal to the cosine value of the angle between the centered arrays. With this conclusion and (1), the relationship between V'' and$\check{V} + \check{A}$is drawn as Fig. 1, which is described as:

$$\begin{cases} |\check{V} + \check{A}|Corr(V + A, W) + \alpha|W| = |\widetilde{V''}|Corr(V'', W) \\ |\check{V} + \check{A}|^2 = |\widetilde{V''}|^2 + (\alpha|\widetilde{V''}|)^2 - 2|\widetilde{V''}|(\alpha|W|)Corr(V'', W) \end{cases}. \tag{8}$$

Thus *Corr(V'', W)* is calculated as:

$$Corr(V'', W) = \frac{|\check{V}+\check{A}|Corr\,(V+A,W)+\alpha|W|}{\sqrt{|\check{V}+\check{A}|^2+(\alpha|W|)^2+2\alpha|W||\check{V}+\check{A}|Corr\,(V+A,W)}} \tag{9}$$

α and $|W|$ are both constant, while V and A are unknown and variable. That is, the distribution of *Corr(V'', W)* depends on the distributions of V and A and is difficult to model.

Frames with different features or through different attacks have r with different distributions more or less. It's ineffective to evaluate r with a uniform T_r.

C. *Improved Extracting Method*

Noticed that$\check{V} + \check{A}$and W are uncorrelated as V and A are both uncorrelated with W, the distribution of angle θ between $\check{V} + \check{A}$and $\check{W}$ conforms the distribution of the angle between two independent random vectors, which is expressed as following [11]:

$$f_\theta(x) = \frac{\Gamma(\frac{N}{2})}{\Gamma(\frac{N-1}{2})} \cdot \frac{\sin^{N-2}(x)}{\sqrt{\pi}}. \tag{10}$$

N is the length of the spread-spectrum code and Γ is the gamma function. Then the probability density function of *Corr(V + A, W)* is calculated as:

$$f_{Corr\,(V+A,W)}(x) = \frac{\Gamma(\frac{N}{2})}{\Gamma(\frac{N-1}{2})} \cdot \frac{(1-x^2)^{\frac{N-3}{2}}}{\sqrt{\pi}}. \tag{11}$$

 ISSN: 2454-4248
76 – 80

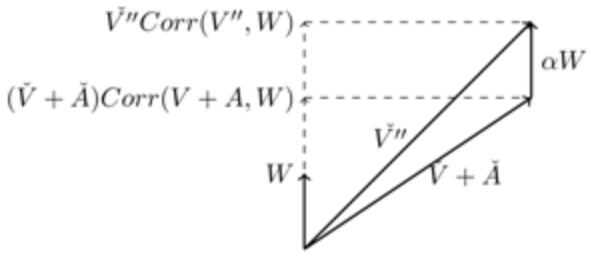

Figure 1. Principle of additive spread-spectrum watermarking.

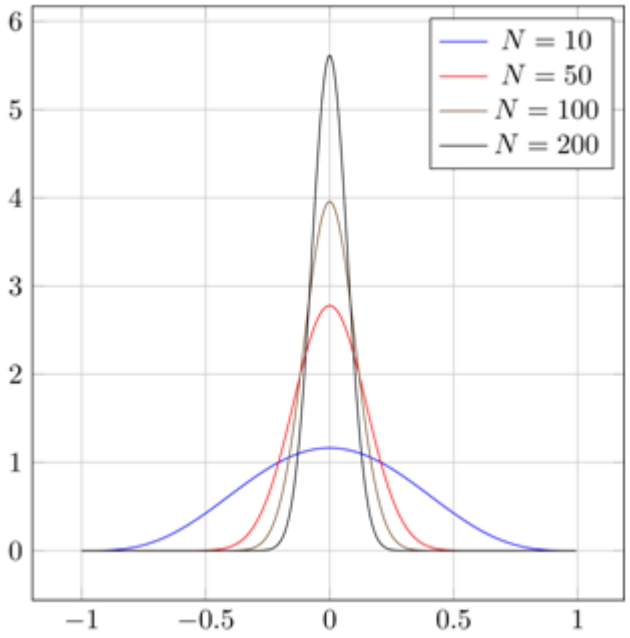

Figure 2. Probability density function of *Corr(V + A, W)*.

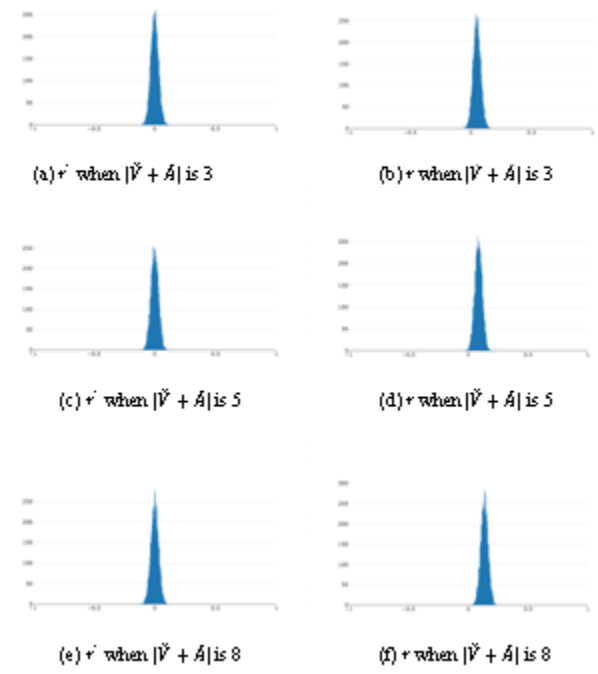

Figure 3. Distributions of r and r' at different $|\check{V} + \check{A}|$.

This is independent of *V* and *A*. Therefore let:

$$r' = Corr(C + A, W) = Corr(C'' - \alpha W, W). \quad (12)$$

The probability that a frame is watermarked can be estimated by the following equation:

$$P_{watermarked} = f_{Corr\,(V+A,W)}(r'). \quad (12)$$

III. Non-Synchronization Multi-Bit Watermark Scheme

Unlike that of r, the distributions of r' in different frames are uniform and invariable with the same W. So we can spread a watermark bit and embed it into every frame in the videos. Then the watermark bit can be extracted by observing the statistical distribution of r' and comparing it with the predicted distribution in (11). As the number of frames observed increases, the result becomes more accurate.

To hide multiple bits in a single video, we can generate different spread-spectrum code for each bit and embed one bit with the corresponding code in each frame. The generated codes should be uncorrelated.

In detail, the embedding process is described as following steps:

1) Generate the uncorrelated spread-spectrum code W_i for each watermark bit ω_i with the secret key K;

2) Randomly choose a watermark bit ω_i to embed;

3) Transform current frame F_t into transform domain coefficients V_t;

4) $V'_t = V_t + \alpha W$ if ω_i=1, otherwise $V'_t = V_t + \alpha(-W)$;

5) Transform V'_t back to watermarked frame F'_t;

6) Switch to the next frame;

7) If βconsecutive frames has been watermarked to hide ω_i, go back to 2), otherwise go back to 3).

When the frame is watermarked with W_i, r' calculated with W_j(i ≠ j) is almost unaffected because of the uncorrelatedness of the spread-spectrum codes.

To extract the watermark bits, we observe the distribution of r' with following steps:

1) Generate the uncorrelated spread-spectrum code W_i for each watermark bit ω_i with the secret key K;

2) Initialize the array γ_i to zeros;

3) Transform the current frame F''_t into transform domain coefficients V''_t;

4) Calculate $r'_{\pm} = Corr(V_t + A_t|_{\pm}, W) = Corr(V''_t \mp \alpha W, W)$ and update $\gamma_i \leftarrow \gamma_i + f_{Corr\,(V+A,W)}(r'_+) - f_{Corr\,(V+A,W)}(r'_-)$ for each watermark bit W_i;

5) Switch to the next frame and go back to 3)

6) If γ_i is larger than 0, it's inferred that ω_i= 1. Or it's inferred that ω_i = 0. The absolute value of γ_i is utilized to measure the reliability of the result.

IV. Results

To prove the performance of our strategy, we have tested it over following transform domain:

1) Given a frame with the resolution of m×n, partition it into 64 × 64 blocks. Each block has a size of $\frac{m}{64} \times \frac{n}{64}$;

2) Calculate the average value of each block to get a matrix with a shape of 64 × 64;

 ISSN: 2454-4248
76 – 80

3) Do wavelet transform on the matrix and hide watermark on the LH, LH and HH coefficients.

This transform is robust to lossy compression attacks and resolution altering attacks.

There are 64 bits embedded into a video with 2000 frames. The watermarked video has a PSNR of 43.5dB with a α of 2. The extracting result without any attacks is demonstrated in Fig. 4. To get unbiased result, the implementation of the strategy of *Hernandez-Avalos et al.* is simplified by assigning a constant function to the feature extractor, which increases the robustness at the expense of the security.

The experiments show that our strategy has a higher BCR, because the proposed non-synchronization strategy replaces the synchronization process with the statistical inference. As a result, an unexpected correlation coefficient of single frame will not affect extractions of later frames.

A. Robustness

The BCR (bit correct ratio) of the proposed algorithm under different type of attacks is demonstrated in Table I:

TABLE I. TABLE TYPE STYLES

Attacks	*Proposed*	*Hernandez-Avalos et al.*
Duplicate 5%	100	78.13
Duplicate 35%	100	78.13
Duplicate 50%	100	78.13
Drop 5%	100	60.94
Drop 35%	100	56.25
Drop 50%	100	53.13
Interchange 5%	100	78.13
Interchange 35%	100	60.94
Interchange 50%	100	60.94
Average	100	75
Reverse	100	50

The Dropping attacks are achieved by dropping sequential frames in every 100 frames. The interchanging attacks are achieved by randomly selecting a frame in the sequence randomly and swapping it with one of the five frames before it or the five frames after it repeatedly. The averaging attack is achieved by simply averaging every two frames into one. And the reversing attack is achieved by evaluating the sequence in opposite order.

B. Analysis

The experimental result shows that the algorithm is robust enough to resist most of the common temporal desynchronization attacks with following reasons:

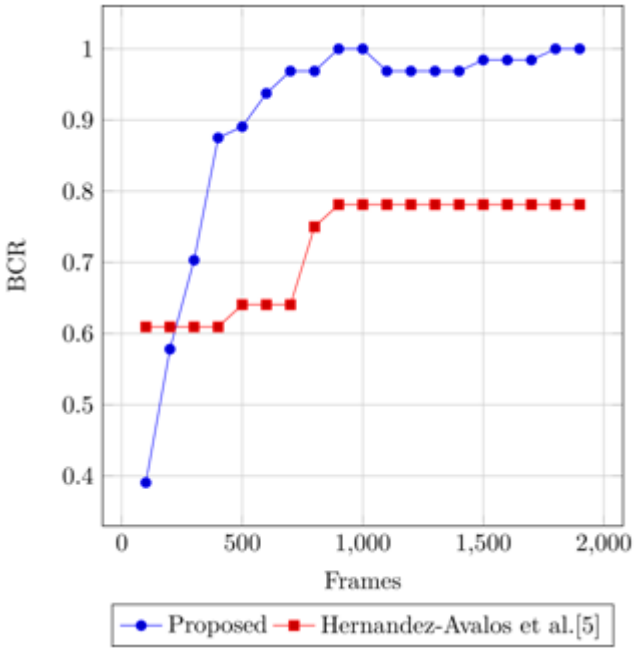

Figure 4. Extracting result.

- The interchange attacks and the reversing attacks change the order of the samples which does not affect the distribution of r'.
- The duplication attacks and the dropping attacks both resample r', but the resampled r' has a similar distribution to which of the original r'.
- The averaging attacks will change the distribution of r' if the two frames averaged are chosen to embed different bits. However, it's negligible as the successive frames tend to embed the same watermark bit.

V. CONCLUSION

This paper develops the probability model of the spread-spectrum watermarking. And then proposes an improved extracting method for spread-spectrum watermarking and a non-synchronization extracting strategy based on statistical inference. The experiments prove its effectiveness and performance.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61373020, Grant U1536102, and Grant U1536116, in part by China Mobile Research Fund Project (MCM20170407).

REFERENCES

[1] F. Hartung and B. Girod, “Watermarking of uncompressedand compressed video,” Signal processing,vol. 66, no. 3, pp. 283–301, 1998.

[2] J. De Cock, H. Hofbauer, T. Stütz, A. Uhl, and A.Unterweger, “An industry-level blu-ray watermarking framework,” Multimedia Tools and Applications,vol. 74, no. 18, pp. 8079–8101, 2015.

[3] C. Chen, J. Ni, and J. Huang, “Temporal statisticbased video watermarking scheme robust againstgeometric attacks and frame dropping,” in InternationalWorkshop on Digital Watermarking, Springer,2009, pp. 81–95.

[4] E. T. Lin and E. J. Delp, “Temporal synchronizationin video watermarking,” IEEE Transactions onSignal Processing, vol. 52, no. 10, pp. 3007–3022,2004.

 ISSN: 2454-4248
76 – 80

[5] P. A. Hernandez-Avalos, C. Feregrino-Uribe, R.Cumplido-Parra, and J. J. Garcia-Hernandez, Videowatermarking method resistant to temporal desynchronizationattacks, US Patent 9,087,377, Jul. 2015.

[6] P. A. Hernandez-Avalos, C. Feregrino-Uribe, R.Cumplido, L. A. Morales-Rosales, C. A. HernándezGracidas,and I. Algredo-Badillo, "Analysis of anadaptive watermarking scheme designed for videocopyright protection," International Journal of ComputerScience and Information Security, vol. 14,no. 12, p. 647, 2016.

[7] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon,"Secure spread spectrum watermarking for multimedia,"IEEE transactions on image processing, vol. 6,no. 12, pp. 1673–1687, 1997.

[8] N. Kokui, H. Kang, K. Iwamura, and I. Echizen,"Best embedding direction for spread spectrumbasedvideo watermarking," in Consumer Electronics,2016 IEEE 5th Global Conference on, IEEE,2016, pp. 1–3.

[9] M. Masoumi, M. Rezaei, and A. B. Hamza, "A blindspatio-temporal data hiding for video ownership verificationin frequency domain," AEU-InternationalJournal of Electronics and Communications, vol. 69,no. 12, pp. 1868–1879, 2015.

[10] S.-K. Ji, W.-H. Kim, H.-U. Jang, S.-M. Mun, andH.-K. Lee, "Robust imperceptible video watermarkingfor mpeg compression and da-ad conversionusing visual masking," in International Workshopon Digital Watermarking, Springer, 2015, pp. 285–298.

[11] Jerkwin. (2014). Probability density distribution of the angle between two random vectors in space. chinese, [Online]. Available: https://jerkwin.github.io/2013/03/18/%E7%A9%BA%E9%97%B4%E4%B8%AD%E4%B8%A4%E9%9A%8F%E6%9C%BA%E5%90%91%E9%87%8F%E9%97%B4%E5%A4%B9%E8%A7%92%E7%9A%84%E6%A6%82%E7%8E%87%E5%AF%86%E5%BA%A6%E5%88%86%E5%B8%83/.