SPREAD SPECTRUM-BASED WATERMARKING FOR TARDOS CODE-BASED FINGERPRINTING FOR H.264/AVC VIDEO

Z. SHAHID, M. CHAUMONT and W. PUECH

LIRMM,UMR CNRS 5506, University of Montpellier II,
161, rue Ada, 34095 Montpellier CEDEX 05, France
zafar.shahid@lirmm.fr, marc.chaumont@lirmm.fr, william.puech@lirmm.fr

ABSTRACT

In this paper, we present a novel approach for active fingerprinting of state of the art video codec H.264/AVC. Tardos probabilistic fingerprinting code is embedded in H.264/AVC video signals using spread spectrum watermarking technique. Different linear and non-linear collusion attacks have been performed in the pixel domain to show the robustness of the proposed approach. The embedding has been performed in the non-zero quantized transformed coefficients (QTCs) while taking into account the reconstruction loop.

Index Terms— Tardos fingerprinting code, active video fingerprinting, H.264/AVC, spread spectrum watermarking.

1. INTRODUCTION

Multimedia content can be easily copied and modified with the evolution of digital media in the recent past and due to it, concerns regarding its protection and authentication have also surfaced. To trace the dishonest users in case of illegal distribution, active fingerprinting (also known as traitor tracing) is used, in which a separate fingerprinting code, identifying a user, is embedded in the personal copy of each user using a robust watermarking technique. To avoid this detection and tracing, the users collude their copies to make a colluded/pirated copy. The collusion process may be of linear or non-linear type. Accusation process extracts the fingerprinting code from the pirated content and aims to trace at least one of the colluders. Theoretically it may not be possible to detect all the colluders, since some of them may have zero contribution in the pirated copy.

In this article, we are presenting the active fingerprinting applied to H.264/AVC video content. H.264/AVC [1] is the state of the art video coding standard of ITU-T and ISO/IEC. It offers better compression as compared to previous video standards. Like previous video standards, an input video frame can be encoded as *intra* or *inter*. In *intra*, spatial prediction is performed while in *inter*, motion compensated prediction is done from previous frames. H.264/AVC video is watermarked off-line in 2^q versions containing different symbols, where q are the bits being embedded in one independent coded unit (called *slice*). We encode a single *intra* frame in more than one *slices* to embed more than 1 bits in it. The online content server just provides the right *slices* according to the user fingerprint sequence. On the decoding side, fingerprint is first extracted, followed by the accusation process accusing some users (or nobody) based on this extracted sequence.

In literature, the designs of these two technologies have been made separately, since both of them have evolved in different research areas. Active fingerprinting of digital content have been mostly studied by the cryptographic community and collusion models are thus defined on the sequence space since the pioneering work [2]. Watermarking has mainly been studied by people in the image or signal processing community. Hence, the effect of collusion of watermarked contents on the fingerprinting codes is not the same as the collusion in the fingerprinting sequence space. In Section 2, previous work on fingerprinting of images is presented. In Section 3, we present the proposed algorithm which includes creation of Tardos fingerprinting codes and its embedding in H.264/AVC video using spread spectrum watermarking technique. Section 4 explains the collusion attacks on fingerprinted video in the pixel domain and its performance analysis. It is followed by concluding remarks in Section 5.

2. PREVIOUS WORK ON FINGERPRINTING CODE

In literature, several fingerprinting algorithms have been proposed for images. In [3], Anti Collusion Code (AAC) was the first anti-collusion forensic code for multimedia content. It was based on combinatorial theories with a joint coding and embedding framework. The code is derived from balanced incomplete block design, which is then modulated by Gaussian orthogonal spreading sequences. In [4], ECC-based forensic code was proposed which uses Gaussian sequences to modulate symbols in the codeword along with additive spread spectrum embedding. There has been some work on designing forensic code for generic data. Boneh and Shaw [2] introduced the concept of marking assumption. Based on this assumption, the Boneh-Shaw code uses a two-level binary code

This work is in part supported by the VOODDO (2008-2011) project of the french Agence Nationale pour la Recherche.

construction to resist collusion. Tardos [5] later proposed an optimal probabilistic binary asymmetric code that reaches the lowest known bound and its q-ary symmetric version is proposed by Skoric *et al.* [6]. In case of multimedia, the colluders usually apply post-processing after collusion. For instance, the colluders can compress the multimedia to reduce the data size to efficiently redistribute the colluded copy. Therefore, it is important to design a collusion resistant forensic code that is robust to post-processing. In [7], Tardos fingerprinting code has been used with zero-bit broken arrows watermarking scheme for images. They have shown that this combination has ruled out the fusion class of attacks. In [8], Lin *et al.* have proposed improved ECC-based anti-collusion codes for images which consume less resources than Tardos fingerprinting codes.

3. PROPOSED ALGORITHM

The proposed scheme is composed of two steps. In the first step, Tardos fingerprinting code is generated. While in the second step, embedding of Tardos fingerprinting code in H.264/AVC video is performed using robust watermarking technique.

3.1. Tardos fingerprinting code generation

For code generation, we have the following three steps:

- For a code of length m, we generate random and independent probabilities $\{p(i)\}_{1\leq i\leq m}$ with the distribution $f(p) = \frac{1}{\pi\sqrt{p(1-p)}}$ for $p \in [0, 1]$. Practically p is between t and $1-t$ with $t = 10^{-3}$. Hence it has high frequency on the edges as shown in Fig. 1.

- The next step is to generate Tardos code. For n users with m code-length, it is a matrix of size $m \times n$ as given in Fig. 2. For the case of binary Tardos code, each line of S is filled with 0 or 1 with $Prob[S(i,j) = 1] = p(i)$. Each column is a fingerprinting code for separate user.

- For accusation process, a sequence Z is extracted from the pirated copy and an accusation score A_j is associated with user j given as:

$$A_j = \sum_{i=1}^{m} U\left(Z(i), S(i,j), p(i)\right), \tag{1}$$

where

$$U(1,1,p) = \sqrt{(1-p)/p} \quad U(1,0,p) = -\sqrt{p/(1-p)},$$
$$U(0,0,p) = \sqrt{p/(1-p)} \quad U(0,1,p) = -\sqrt{(1-p)/p}.$$

A fingerprinting code is analyzed based on its code-length, maximum number of colluders c_0, false-positive (ϵ_1) and false-negative (ϵ_2) values. For binary asymmetric Tardos code, the length of code is given as $m = 100c_0^2 \ln\left(\frac{1}{\epsilon_1}\right)$ and ϵ_1 and ϵ_2 is given as $\epsilon_2 = \epsilon_1^{\frac{c_0}{4}}$ [5]. The code-length is further reduced in its symmetric version by Skoric *et al.* [6] and both ϵ_1 and ϵ_2 were made independent of each other.

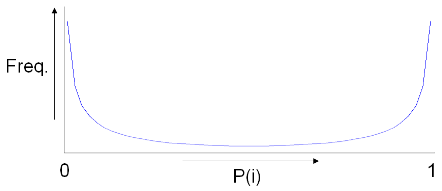

Fig. 1. Probability distribution for the probabilities of Tardos code.

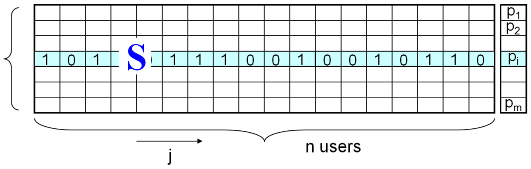

Fig. 2. $m \times n$ Tardos code with code-length of m for n users.

3.2. Spread Spectrum Strategy

Many robust watermarking techniques exist in literature. Spread spectrum watermarking technique offers robustness [9] and capacity [10] and has been selected for our investigation. Spread spectrum embedding is resistant against number of attacks. It was argued to be highly resistant to collusion attacks, when the watermarks have a component-wise Gaussian distribution and are statistically independent [11]. The basic intuition of this natural strategy is that the randomness inherent in such watermarks makes the probability of accusing an innocent user very unlikely. Spread spectrum embeds the watermark in overlapped regions and this spreading makes it challenging to change even a single bit at will. This confines the effect of a colluder's action to a milder form of collusion from the designer's point of view. Let $S(i,j)$ be the i^{th} bit of Tardos fingerprinting code S_j which is to be embedded into a block of host vector X. To increase the energy of the embedding bit, we specify a scaling parameter α, which is decided based on the human perception. So the watermarked block is given as:

$$Y = X + \alpha U_i (-1)^{S(i,j)}, \tag{2}$$

with $S(i,j) = 0$ or 1 and U_i is a Gaussian sequence. The attacked watermarked signal is $Z = Y + n$, where n is the noise due to attack. The watermark bit $\tilde{S}(i,j)$ is extracted from Z by the linear correlation of Z and U_i of length l as:

$$\tilde{S}(i,j) = \begin{cases} 0\,, & \text{if } \sum_{j=1}^{l} Z[j]U_i[j] > 0 \\ 1\,, & \text{if } \sum_{j=1}^{l} Z[j]U_i[j] < 0. \end{cases} \tag{3}$$

3.3. Embedding strategy

For embedding a Tardos code in QTCs of H.264/AVC, embedding can be done in entropy coding stage. It is analogous to embedding watermark in a compressed bitstream. This includes two watermarking approaches. The first approach embeds watermark in VLC domain and bitstream is only to be entropy decoded to use this e.g. as proposed by Lu *et al.* [12]. Another approach embeds watermark in DCT domain and for this approach, bitstream has to be entropy decoded and inverse quantized e.g. differential energy watermarking [13]. Embedding watermark, after reconstruction loop, creates two problems. First, we do reconstruction with QTC on the encoder side, while on the decoder side with watermarked QTC. This results in a mismatch on the decoder side, which keeps on increasing because of the prediction process and loss in PSNR is very significant even for *intra* frames. Second, Rate Distortion (RD) bit allocation algorithm works in quantization module and any change in bitrate/quality trade-off because of the watermarking of QTCs is not taken into account.

To solve both problems, watermark embedding should be performed inside the reconstruction loop as shown in Fig. 3. In this case, we have the same watermarked QTC $\hat{y}_w(u,v)$ on both encoder and decoder side for prediction. In this case, RD bit allocation algorithm is working on $\hat{y}_w(u,v)$ for both *intra* and *inter* frames. For embedding of m bits of Tardos code S_j, the content is divided into m blocks and 1 bit is embedded into each block. It is important to note that every block should contain at least one slice, if it is lesser than one video frame. For larger multimedia content, we can have slices of larger size and hence the embedding will be more robust. In $intra$ 4×4 mode, scanning of 4×4 blocks inside MB is not in a raster scan fashion. So we will create a vector X for spread spectrum while taking this scan into account. In our case, X consists of DC transform coefficients of 4×4 transform blocks. Each bit of Tardos Codes is embedded into the host vector X using spread spectrum insertion as illustrated in Fig. 4.

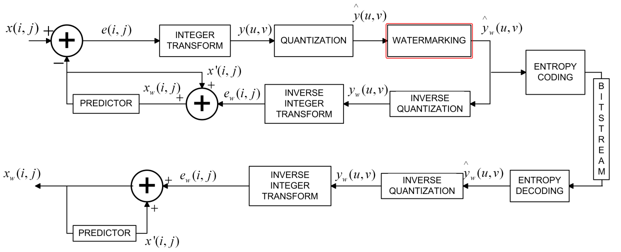

Fig. 3. Watermark embedding in H.264/AVC while taking into account the reconstruction loop.

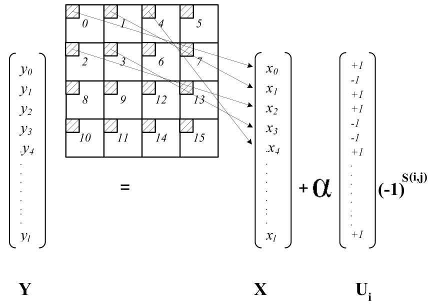

Fig. 4. Spread spectrum embedding of $S(i,j)$ Tardos code bit in copy of user j.

4. EXPERIMENTAL SIMULATION

We have used the reference implementation of H.264 JSVM 10.2 in AVC mode for video sequences in CIF resolution. We have selected $intra4 \times 4$ MB mode for encoding *intra* sequence along with CAVLC at QP value 18 and have embedded fingerprinting code in both *luma* and *chroma*. For the experimental results, nine benchmark video sequences, which includes 'bus', 'city', 'foreman', 'football', 'soccer', 'harbour', 'ice' and 'mobile', have been concatenated in a repeated fashion for this simulation.

For generation of binary Tardos code, the parameter values are: $n = 100$, $\epsilon_1 = 10^{-3}$, $c_0 = 20$ and $m = 92104$ [1]. 10 bits of Tardos code were embedded per frame using robust spread spectrum watermarking. Hence we used 9211 frames (369 seconds of video at 25 fps) of CIF resolution to embed the code. For accusation process, an accusation sum A_j is calculated for each user j as explained in Section 3.1. A_j for accused users may be modeled with a Gaussian centered at $\mu = \frac{2m}{c\pi}$, while A_j for innocent users may be modeled with a Gaussian centered at 0. Accused users (the traitors) have a score above $\mu - \sqrt{m}$ (*i.e.* $\frac{2m}{c\pi} - \sqrt{m}$), where $\sqrt{m}$ is the standard deviation of the Gaussian. A more precise threshold may be selected such that proposed in [14][2].

4.1. Collusion attacks

In this work, linear and non-linear collusion attacks have been performed in the spatial domain. Linear attacks include averaging attack. Under this attacks, each pixel in pirated video is average of the corresponding pixels of the fingerprinted videos associated with the colluders. The typical nonlinear collusion attacks are minimum/maximum/median attacks, minmax attack and modified negative attack. For minimum/maximum/median attacks, each pixel in pirated video is the minimum, maximum, or median, of the corresponding pixels of the fingerprinted videos. For minmax attack, each colluded pixel is the average of the maximum and minimum of the corresponding pixels of the fingerprinted videos. For modified negative attack, each pixel of the attacked video is the difference between the median and the sum of the maximum and minimum of the corresponding pixels of the fingerprinted video signals. Let S_c is a set of all

[1] Normally, in traitor tracing, $\epsilon_1 = 10^{-6}$. To reduce code-length and hence, the simulation time, we have selected $\epsilon_1 = 10^{-3}$.

[2] It links ϵ_1 and threshold.

K colluders, the collusion functions can be given as:

$$Z_{ave}(j) = \sum_{k \in S_c} \frac{Y_k(j)}{K} \tag{4}$$

$$Z_{min}(j) = \min\{Y_k(j)\}_{k \in S_c} \tag{5}$$

$$Z_{max}(j) = \max\{Y_k(j)\}_{k \in S_c} \tag{6}$$

$$Z_{med}(j) = med\{Y_k(j)\}_{k \in S_c} \tag{7}$$

$$Z_{minMax}(j) = (Y_{min}(j) + Y_{max}(j))/2 \tag{8}$$

$$Z_{modNeg}(j) = Y_{min}(j) + Y_{max}(j) - Y_{med}(j) \tag{9}$$

where Z is the colluded pixel. Fig. 5 illustrates the PSNR of the colluded video, created using linear and non-linear attacks. It also shows the PSNR of original copy. In case of average/median/minmax attacks, PSNR gets better with increase in number of colluders. While for minimum/maximum attack, the quality of video slightly decreases as the number of colluders increases.

Table 1 shows the number of colluders which have been successfully traced by analyzing a pirated video content. In most of the cases, the colluders have been successfully traced. The attack which makes the video non-traceable also degrades the quality of the video very badly. For example, in case of modified negative attack, we could detect only few of the colluders, but PSNR of the attacked video is 15 dB. While on the other hand, averaging attack does not degrade the visual quality rather PSNR gets better in some cases but we can also detect the colluders with very high confidence value. Among the non-linear attacks, modified negative attack makes the accusation process difficult but it severely degrades the video quality.

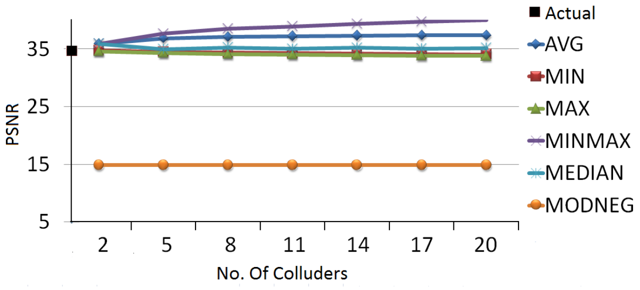

Fig. 5. PSNR of colluded video content, which has been generated using collusion attacks.

	Number of colluders detected for attacks					
K	avg	min	max	median	minMax	modNeg
2	2	2	2	2	2	2
5	5	5	5	5	5	5
8	8	8	8	8	8	6
11	11	10	10	10	10	7
14	14	13	13	13	13	9
17	16	15	16	16	16	10
20	18	18	18	19	18	11

Table 1. Number of colluders traced from the K-colluded copy, generated using different collusion attacks.

5. CONCLUSION

In this work, we have demonstrated the fingerprinting of H.264/AVC using Tardos Fingerprinting codes. Spread spectrum robust watermarking technique has been used for embedding of fingerprinting code in personal copies. Our experimental analysis verifies that embedding of Tardos codes using spread spectrum watermarking technique is a very good design indeed. The colluded video content can be successfully analyzed to trace the colluders until its quality becomes unacceptable. In future, we intend to extend our work for both *intra* and *inter* frames for all of their MB modes, while incorporating informed watermarking technique.

6. REFERENCES

[1] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)," Tech. Rep., Joint Video Team (JVT), Doc. JVT-G050, March 2003.

[2] D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, Sep 1998.

[3] W. Trappe, M. Wu, Z. Wang, and L. Liu, "Anti-Collusion Fingerprinting for Multimedia," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1069–1087, 2003.

[4] S. He and M. Wu, "Joint Coding and Embedding Techniques for Multimedia Fingerprinting," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 231–247, June 2006.

[5] Gábor Tardos, "Optimal Probabilistic Fingerprint Codes," in *Proc. ACM symposium on Theory of computing*, New York, NY, USA, 2003, pp. 116–125.

[6] B. Skoric, S. Katzenbeisser, and M. Celik, "Symmetric Tardos Fingerprinting Codes for Arbitrary Alphabet Sizes," *Designs, Codes and Cryptography*, vol. 46, pp. 137–166, 2008.

[7] F. Xie, T. Furon, and C. Fontaine, "On-Off Keying Modulation and Tardos Fingerprinting," in *Proc. ACM workshop on Multimedia and security*, New York, NY, USA, 2008, pp. 101–106.

[8] W. Lin, S. He, and J. Bloom, "Performance Study and Improvement on ECC-Based Binary Anti-Collusion Forensic Code for Multimedia," in *Proc. ACM workshop on Multimedia and security*, New York, NY, USA, 2009, pp. 93–98.

[9] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, vol. 6, pp. 1673–1687, 1997.

[10] P. Moulin and J. O'Sullivan, "Information-Theoretic Analysis of Information Hiding," *IEEE Transactions on Information Theory*, vol. 49, pp. 563–593, 2003.

[11] F. Hartung, J. Su, and B. Girod, "Spread Spectrum Watermarking: Malicious Attacks and Counterattacks," in *Proc. SPIE: Security and Watermarking of Multimedia Contents*, 1999, pp. 147–158.

[12] C. Lu, J. Chen, and K. Fan, "Real-Time Frame-Dependent Video Watermarking in VLC Domain," *Signal Processing: Image Communication*, vol. 20, pp. 624 – 642, 2005.

[13] G. Langelaar and R. Lagendijk, "Optimal Differential Energy Watermarking of DCT Encoded Images and Video," *IEEE Transactions on Image Processing*, vol. 10, pp. 148–158, 2001.

[14] F. Cérou, T. Furon, and A. Guyader, “Experimental Assessment of the Reliability for Watermarking and Fingerprinting Schemes,” *EURASIP Journal on Information Security*, pp. 1–12, 2008.