Ensuring security of H.264 videos by using watermarking

Marc Chaumont

Ensuring security of H.264 videos by using watermarking

Marc CHAUMONT

University of Nîmes, Place Gabriel Péri, 30000 Nîmes, France.
Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II,
161, rue Ada, 34392 Montpellier cedex 05, France.

ABSTRACT

Watermarking is known to be a very difficult task. Robustness, Distortion, Payload, Security, Complexity are many constraints to deal with. When applied to a video stream, the difficulty seems to be growing in comparison to image watermarking. Many classical non malicious manipulations of a compressed stream may suppress the embedded information. For example, a simple re-compression of a DVD movie (MPEG2 compression) to a DivX movie will defeat most of the current state-of-the-art watermarking systems. In this talk, we will expose the different techniques in order to watermark a video compressed stream. Before, we will present the H.264/AVC standard which is one of the most powerful video-compression algorithms. The discussion about video watermarking will be illustrated with H.264 streams. The specific example of traitor tracing will be presented. Deadlocks will be discussed, and then the possible extensions and the future applications will conclude the presentation.

Keywords: State of the art, Tutorial, Watermarking, H.264, Joint watermarking and video compression, Security applications, Traitor Tracing

1. INTRODUCTION

In our everyday's life, we are generating, visualizing, storing, or/and editing video movies. All those videos represent a huge quantity of information. The network bandwidth and the storing devices are limited. There is thus a strong need for compressing those data. As an example, 90 minutes of a Standard Definition TV movie at 25 frames per second necessitate, without compression, a rate of 237 Mbits/s and a storing capacity of 1.22 Tera-bits. Compression is required in order to stream the video on an ADSL connexion at 20 Mbits/s (it costs around 30 Euros by month in France) or in order to store the movie on a DVD whose capacity is 4.7 GBytes.

The compression community works in order to provide compression standards since 25 years.[1] Currently, the state-of-the-art codec is H.264/MPEG4-Part10.[2] The producers, the distributors (Hollywood...), the cinema operators, the technology providers (Internet providers, reading and recording devices constructors, ...) are very dynamic in the standardizations process. Nowadays, one of their challenges is to provide additional features such that security solutions in order to protect their rights. For example, they need to protect themselves from pirating. There is many possible solution in order to propose some security such that using cryptography, secure all the devices from a reader to a displayer (Example: Blue-Ray reader + HD TV + HDMI wire), spy the network and collect pirate IPs, propose cheap movies renting, watermarking...

In this tutorial, we will talk about security with watermarking schemes integrated inside an H.264 codec. We will first recall few principles about H.264 video coding. We then recall the robust video watermarking principles and give three concrete examples. The security notion is then explained. We finally conclude the talk with an example of a traitor tracing application based on a watermarking scheme integrated to H.264.

Send correspondence to Marc.Chaumont@lirmm.fr.

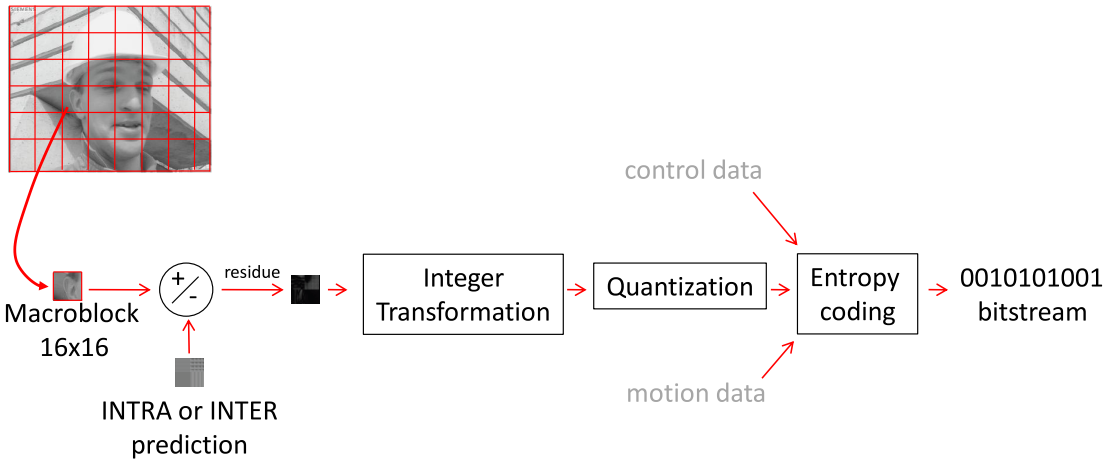

Figure 1. General H.264 coding scheme.

2. A BRIEF INTRODUCTION TO H.264

H.264 or MPEG-4 Part 10[2] is the state-of-the-art for video coding*. The first version of the standard was approved in 2003. The standard comes from two organizations: ITU-T and ISO/IEC. The H.264 encoding allows for up to 50% throughput improvement compared to MPEG2 or MPEG4 Part 2 simple profile.[1] Since the MPEG1 normalization in 1988, the encoding mechanism is based on: the decomposition of the image into blocks, the spatial or temporal prediction, and the quantization and coding of coefficients. MPEG1, MPEG2, MPEG4-Part2, and H.264 are usually named block-based encoders. To simplify the explanation, we give a simplified diagram in Figure 1, showing some important steps.

The current image to be encoded is divided into macroblocks of size 16×16. Each macro block consists of a variable number of blocks that are temporally or spatially predicted depending on whether the frame is Inter or Intra. A 4×4 block is predicted using one of the prediction methods, and then the residue block is computed. The residue block corresponds to the difference between the current block and the prediction block. It then undergoes a frequency transformation (the integer transformation). The resulting block is then quantized and coefficients are entropy coded by a CABAC or a CAVLC coder.

In the case of a temporal prediction, the residue block is also computed and then the transformation, the quantization, and the encoding are substantially the same. When there is a temporal prediction (case of Inter frames) the temporal prediction consists to determine in a frame already coded (temporally preceding or following) the nearest block in absolute distance or L2 distance. The position of the nearest block is then represented by a displacement vector relatively to the current block to predict. This displacement vector (motion vector) is also encoded.

Let us now give a brief introduction to robust watermarking.

3. FEW WORDS ABOUT THE WATERMARKING

Robust watermarking is the art of modifying a media (image, sound, video ...) such that:

- it contains a message most of the time in relation with the media,
- the degradation is imperceptible,
- the hidden message is not lost when media degradation occurs (attacks).

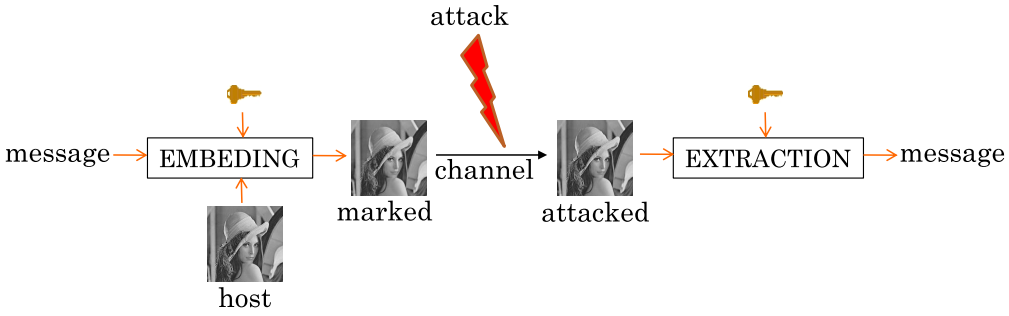

Figure 2. General watermarking scheme.

The figure 2 show the general scheme of a watermarking system (with a blind extraction). A message is secretly embedded in a host image in order to obtain a marked image. The marked image is then distributed to someone who may degrade it maliciously or not (the degradation is named an attack). The secret key owner may try to extract/detect the message from the marked-attacked image. The scheme is robust if the secret key owner is able to extract the message without any bit errors. As an illustration of an attack to the robustness, Figure 3.a show the Lena image which have been watermarked with the 0-bit Broken Arrows algorithm[3] and Figure 3.b show the image attack by the addition of a white Gaussian noise. The watermark is still detected in the attacked image.

In the case of video watermarking, the number of possible attacks where the scheme should be robust is very huge. The table 1 lists few of them.[4] Remark that the spatial desynchronization and the temporal desynchronization are really difficult to manage. The desynchronization problem has been strongly studied in the years 2000 but there are still lots of work in order to obtain satisfying solutions. The temporal desynchronization problem is also difficult to manage and there are very few solutions.

*The groups of experts from ITU-T (Video Coding Experts Group: VCEG) and ISO/IEC (Moving Picture Experts Group: MPEG) initiated the standardization of the next encoder generation named HEVC (High Efficiency Video Coding). The standardization is scheduled for July 2012 and the associated documentation may be obtained at http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx.

(a) Watermarked image (b) Additive White Gaussian Noise Attack

Figure 3. Illustration of the robustness of Broken Arrows (BA) algorithm.[3]

Table 1. List of few non-malicious attacks.[4]

Photometric	Noise addition, DA/AD conversion Gamma correction Transcoding and video format conversion Intra and inter-frames filtering Chrominance resampling (4:4:4, 4:2:2, 4:2:0)
Spatial Desynchronization	Changes display formats (4/3, 16/9, 2.11/1) Changes resolution (NTSC, PAL, SECAM) Positional jitter Hand-held camera recording (curved-bilinear transform)
Temporal Desynchronization	Changes of frame rate Frame dropping / insertion Frame decimation / duplication
Video editing	Cut-and-splice and cut-insert-splice Fade-and-dissolve and wipe-and-matte Graphic overlay (subtitles, logo)

4. WATERMARKING IN A VIDEO BITSTREAM

To facilitate the classification of the different approaches, we can classify them based on the place where the watermarking occurs during the video compression. The four places are: before compression, before quantization, after quantization and during entropy coding. The table 2 gives some references on representatives of the different approaches (the list is not exhaustive).

Table 2. Classification of few video watermarking approaches.

Before compression	Inside H.264 structure		
	During encoding process	After quantization	During Entropy coding
Images	**Luma modification:**	During encoding process	Mobasseri and Raikar, 2007[5]
SS: Cox *et al.*, 1997[6]	Golikeri *et al.*, 2007[7]	Shahid *et al.*, 2009[8]	Zou and Bloom, 2009[9]
DPTC: Miller *et al.*, 2004[10]	**Motion vectors modification:**	Noorkami and Mersereau, 2008[11]	...
P-QIM: Li and Cox, 2007[12]	Zhang *et al.*, 2001[13]	...	
...	**GOP structure modification:**	In an already	
Sequence of images	Linnartz and Talstra, 1998[14]	H.264 encoded bitstream	
Temporal watermarking:	...	Hartung and Girod, 1998[15]	
Haitsma and Kalker, 2001[16]		Gong and Lu, 2008[17]	
3D DFT:		...	
Deguillaume *et al.*, 1999[18]			
On-off keying (BA):			
Xie *et al.*, 2008[19]			
...			

Before compression we find images watermarking approaches such as spread spectrum,[6] DPTC,[10] or P-QIM.[12] There are also approaches that take into account the temporal dimension such as the watermarking in the histogram average frames luminance[16,20] or the watermarking on a frequency volume such as the 3D-DFT.[18]

Before the quantization phase, one may modify the luminance DCT coefficients, taking care to have an embedding strength parameter which is function of the quantization parameter chosen by the user. We can cite the clever proposal of Golikeri *et al.*[7] that embeds 1 bit per macro block, uses a psychovisual mask, and combines a spreading and a quantization watermarking technique (the watermarking technique is the ST-SCS

quantization approach). This approach is also one of the most robust approaches integrated into an H.264 video stream. One can also embed the watermark signal in the motion vectors.[13] Finally, Linnartz and Talstra[14] proposed a method modifying the structure of a GOP (Group Of Picture) by varying the number of IPB frames within a GOP. The latter method is not robust to a re-encoding.

After the quantization phase, we can distinguish two different problems: either the embedding is performed on a stream to be encoded, either the embedding is performed on a stream already encoded. For an embedding in a stream to be encoded, Noorkami and Mersereau[11] propose a robust 0-bit approach, using a psychovisual mask, with an embedding in the ACs coefficients and a detection that can be performed without knowledge of the positions of the watermarked coefficients. We have also proposed a method that takes place after the quantization phase[21] and takes into account the compression performances using the rate-distortion optimization. For an embedding performed on an already compressed stream, there is little room to make changes. We can cite for example[15] and.[17] Those kind of approaches are very interesting in the case of a video streaming application where the watermarking signal is integrated in an already compressed movie.

Finally, one may achieve the watermarking during the entropy coding phase. This kind of approach necessitates to enter the very complex structure of CABAC or CAVLC. Mobasseri and Raikar,[5] Zou and Bloom[9] among others have proposed this kind of approach. The payload is obviously very low and approaches are not robust. In return, the bit rate of the resulting file is not modified. Also note that some approaches tend to generate a drift between what was coded and what will be decoded.

To conclude with this rapid state-of-the-art of video watermarking the good news is that there are good solutions robust to photometric attacks inside H.264 or a similar codec. The bad news is that most of the solutions (all?) inside H.264, or a similar codec, are not robust, or not enough robust, to temporal and spatial desynchronizations. If we are looking to a solution that is robust to desynchronizations, one probably has to achieve the watermarking outside of a video codec.

Now, an additional problem in watermarking and thus in video watermarking is to take care to the security aspects. This is the object of the next section.

5. WATERMARKING AND SECURITY

5.1 Few word about security

The classical framework of security in watermarking is really different from the robustness. Kerckhoffs's framework[22] is such that:

- The embedding and extracting algorithms are known by the attacker,
- the only secret parameter is the key,
- and the attacker owns observations.

A security attack is an attack for which secrets parameters or secret information are obtained (whereas a robustness attack only tries to destroy or suppress the watermark signal). Security problem addresses the analysis and the construction of secure algorithms, and the analysis and the construction of security attacks.[23,24]

Recently, few practical security attacks have been achieved for images for different watermarking categories. The table 3 lists some of them. The security problem for video has not been studied since the formal analysis of.[23,24] We may find a quiet old but really interesting study of video watermarking security in.[25] The general principle of the two type of attacks (collusion type I and II) are presented and the two main security applications are evoked: the copyright application and the traitor tracing application. Note that the video security is more difficult to ensure than for image watermarking, since there is many more observation (there is more images). Moreover, video security attacks may be achieved thanks to collusion (using of a group of images) of several videos (Inter collusion) or collusion inside the same video (Intra collusion). Note that the Intra collusion attack is a really strong security challenge since an attacker does not have to obtain many videos or many versions of the video to achieve an attack.

In the next section we propose a practical example of a traitor tracing application integrated inside an H.264 stream.

Table 3. List of few practical security attacks on image watermarking schemes.

Images	Proposed attacks
Spread Spectrum	"Comparison of Secure Spread-Spectrum Modulations Applied to Still Image watermarking", B. Mathon, P. Bas, F. Cayre, and B. Macq, Annals of Telecommunication, 2009.[26]
Broken Arrows	"Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme", P. Bas and A. Westfeld, MM&Sec'2009.[27] Counter Attack: "Better Security Levels for 'Broken Arrows' ", F. Xie, T. Furon, and C. Fontaine, SPIE'2010.[28]
DPTC	"Evaluation of an Optimal Watermark Tampering Attack Against Dirty Paper Trellis Schemes", P Bas and G. Doërr, MM&Sec'2008.[29]
Quantized based	"Exploiting Security Holes in Lattice Data Hiding", L. Pérez-Freire and F. Pérez-Gonzàlez. IH'2007.[30]

5.2 A traitor tracing application

The objective of traitor tracing is to deter the illegal use of a product. After a transaction between a seller and a/some buyer(s), the traitor tracing technique allows the seller to retrieve the buyer(s) that redistributes the product illegally. When selling a film, the seller includes a code word which will identify the buyer if needed. If the seller discovers that his film(s) is (are) redistributed unlawfully, the traitor tracing technique should enable him to identify the buyer(s) responsible for this redistribution.

The traitor tracing approach must provide a safe mechanism that allows the seller to engage penalties or lawsuits against the fraudster(s) (we call them traitors). The traitor tracing mechanism must provide an accusation's mechanism in which no innocent person can be accused. In particular, the traitor tracing mechanism must yield impossible the identity usurpation. The traitor tracing mechanism should also identify at least one traitor if there are illegal redistributions of copies built from the traitors versions.

In 2003, Tardos[31] proposed a probabilistic code adapted to the problem of traitor tracing. Code length was estimated at $m = 100c^2 ln(\frac{n}{\epsilon_1})$ given n users, c colluders and with ϵ_1 the probability of accusing an innocent. Note that the constant 100 in the code length was subsequently reduced[32–34] depending on different assumptions... All the optimizations proposed in[32–34] have carry on the Tardos work[31] and optimized the parameters without fundamentally changing the code or the accusation algorithm.

The generation of a Tardos code of dimension m is extremely simple and involves two steps:

- **Generation of the probability values**: Initialize a pseudorandom generator with a secret key and then generate m real values between 0 and 1 whose distribution is $f(p) = \frac{1}{\pi\sqrt{p(1-p)}}$ with $p \in [0, 1]$. Those m values are noted $\{p(i)\}_{1 \leq i \leq m}$ and will correspond to the probability of having 1 for the ith bit of a code word;
- **Generation of the Tardos code**: For n users, we generate a matrix $\mathbf{S}$ of size $m \times n$. A column represents a code word associated with a user. The rows of the matrix are filled with 0 and 1 such that $Prob[\mathbf{S}(i, j) = 1] = p(i)$.

The accusation process is extremely simple since there is only to compute an accusation score, A_j, which is computed from the extracted word $\mathbf{z}$ and the code word from user j:

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

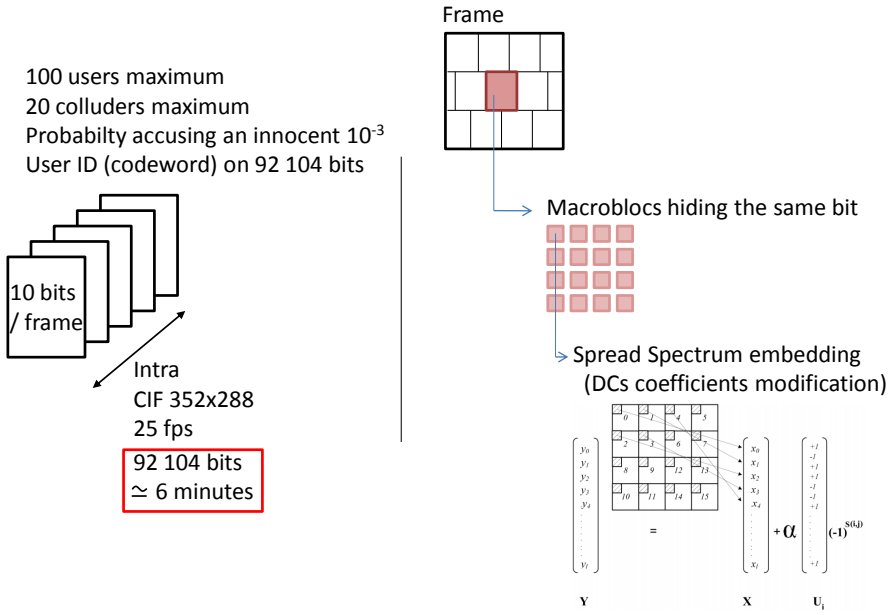

Figure 4. Watermarking of a Tardos codeword in H.264 in Shahid *et al.*[21]

with U the function defined for $p \in [0,1]$ by:

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

If the score is above a given threshold, the user j is accused.

We now studies the integration feasibility of a traitor tracing within an H.264 video. This approach has been published ICIP 2010 by Shahid *et al.*[21]

First, each frame is cut into spatial areas in order to embed 1 bit in each areas. Conveniently, each area is stored in a *slice*. A slice is a part of the binary stream (it is a portion of the compressed file) that can be easily accessed and whose content is decodable independently of the other slices. Figure 4 shows the cutting of a 352×288 frame. The frame is split in 10 areas (slices), each one embedding 1 bit[†]. Note that the watermarking process can be done offline. For each slice, two different versions are generated: a slice embedding the value 0 and the other embedding the value 1. When a video is distributed to a buyer, the seller assigns a Tardos code word to the buyer, and then built the video using the already watermarked slices.

The watermarking principle is very simple. Once the slicing cut is made, the watermarking is achieved by the spread spectrum technique (1 bit is embedded in 1 slice). The slice is made of several macro blocks. The macro blocks from the same slice will be modified so that they embed the same binary value. The watermarking principle is the same for all the macro blocks. The quantized DC coefficients from the blocks of a macro block are stored in a host vector $\mathbf{x}$. We then apply the spread spectrum[‡] (see Equation 2). We generate a secret carrier $\mathbf{u}_i \in \{-1,1\}^n$ (with n the host vector size), we modulate the ith bit of the Tardos code word $\mathbf{S}(i,j)$ of the jth user to get -1 if $\mathbf{S}(i,j) = 1$ and $+1$ if $\mathbf{S}(i,j) = 0$, then we spread the modulated bit on the carrier (multiplication of the carrier and the modulated bit) to get the watermark vector. The watermark vector is then amplified or reduced by multiplying it by a strength factor (α scalar), and finally we add it to the host vector $\mathbf{x}$ to obtain

[†]In our experiments,[21] we set the code length to $m = 20.c^2.ln(\frac{n}{\epsilon_1})$.

[‡]Note: The spread spectrum technique is acutely integrated into the H.264 encoder during the compression phase. When a block is being encoded, the watermarking is achieved for all the possible prediction modes, and then the coder selects the prediction mode of the block which will give the best rate-distortion tradeoff.

the watermarked vector $\mathbf{y}$:

$$\mathbf{y} = \mathbf{x} + \alpha.\mathbf{u}_i.(-1)^{\mathbf{S}(i,j)} \tag{2}$$

The vector $\mathbf{y}$ corresponds to the new quantized DC coefficients. It should be noted that an approach that modifies the DC coefficients produces unpleasant blocks effects. We can reasonably think that the use of a robust psychovisual mask and an error correcting code, taking into account erasure, may solve this problem.

During the extraction step, for each macro block, the DC coefficients are re-extracted to form the watermarked-attacked vector $\mathbf{z}$. The correlation between the carrier $\mathbf{u}_i$ and the vector $\mathbf{z}$ gives us the embedded bit (and thus the ith bit of the codeword assigned to the jth user; this bit is noted $\tilde{\mathbf{S}}(i,j)$):

$$\tilde{\mathbf{S}}(i,j) = \begin{cases} 0, & \text{if } \sum_{k=0}^{n} \mathbf{z}[k].\mathbf{u}_i[k] > 0 \\ 1, & \text{if } \sum_{k=0}^{n} \mathbf{z}[k].\mathbf{u}_i[k] < 0. \end{cases} \tag{3}$$

To evaluate the scheme, only the collusion attacks, between different versions of the same movie, have been tested. The robustness of the scheme (to photometric and valumetric attacks) and also its security (facing intra collusion attacks) will have to be studied in the future. However, the primary objective of this study was to evaluate the feasibility of a traitor tracing scheme using a Tardos code but also to propose a watermarking solution integrated to a compression scheme. In that sense, the results are promising.

The collusion attacks are all carried out in the spatial domain. All the traitors have a different version of the same video and they forge a new video using a particular strategy. Each pixel of the new video is obtained using one of the following strategies: computation of the average, computation of the minimum, computation of the maximum, computation of the median, computation of the minmax (average between maximum and minimum) and computation of the mod-neg (minimum plus maximum minus median).

We use the JSVM 10.2 implementation of H.264 with videos in CIF resolution (352x288). We choose the 4x4 Intra mode for Intra coding, the CAVLC encoder, set the quantization to QP = 18, and embed the Tardos code words in both the luminance and chrominance. Nine sequences ('bus', 'city', 'foreman', 'football', 'soccer', 'harbour', 'ice, 'mobile', 'crew') are concatenated and repeated to form the video to be watermarked. The parameters of the Tardos code are n = 100 users, $\epsilon_1 = 10^{-3}$, $c = 20$ colluders. The code length is thus $m = 92104$. We embed 10 bits per frame. 9211 frames are required in order to completely embed a Tardos code word. It corresponds to about 6 minutes of a video at 25 frames per second.

Table 4. List of few practical security attacks on image watermarking schemes.

	Number of colluder detected for each attack strategy					
Number of colluder	avg	min	max	median	minmax	mod-neg
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

The PSNR of the watermarked videos is 35 dB and all attack strategies lead to a PSNR close to or even greater than 35 dB for collusion from 1 to 20 colluders. Only the mod-neg strategy lead to a 15 dB PSNR. The latter strategy is unattractive because the video quality is so low that pirated video becomes useless. Table 4 shows the number of colluders who were successfully detected in a pirated video, for different strategies of collusion and a variable number of colluders. The most effective strategy is the mod-neg attack but it is also an unuseful attack. By cons, for all the other attacks, no matter the number of traitors, the accusation process determines almost all the traitors. Knowing that in a traitors tracing application, the most important thing is to determine at least one traitor, the implementation approach works extremely well.

This work highlighted the feasibility of a traitors tracing system with a watermarking approach integrated in an H.264 video. Tardos code was used to trace a traitor in a video obtained by collusion. The watermarking system and the code have been designed to be realistic in the case of a very small number of users (100). None of the collusion strategy has put in default the accusation process.

6. CONCLUSION

This tutorial recall the H.264 important steps and gives some pointers about video watermarking. Then, the security notions are recall and some security attacks for image watermarking schemes are evoked. The video watermarking security is a step beyond in difficulty since there is lots of observation that may be used to build an attack. We then present a proposition[21] for a traitor tracing integrated in an H.264 coder. It is obvious that the integration of the traitor tracing system in a compressed video is not yet a completely solved problem. That said, the proposition[21] works well and helps highlight the work that remains to be done to get more mature systems. It is clear that the length of the anti-collusion codes must be reduced. The robustness and the security of video watermarking systems should be improved and re-evaluated with recent knowledge in security and informed watermarking. Finally, video watermarking schemes are rarely robust to desynchronization attacks and this should be take into account in the future.

REFERENCES

1. Richardson, I., [*The H.264 Advanced Video Compression Standard*], Wiley, 2nd ed. (2010).
2. "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) (2003).
3. Furon, T. and Bas, P., "Broken Arrows," *EURASIP Journal on Information Security* **2008**, 3:1–3:13 (2008).
4. Doërr, G., *Security issue and collusion attacks in video watermarking*, PhD thesis, University of Nice-Sophia Antipolis, France (June 2005). Supervised by J.-L. Dugelay.
5. Mobasseri, B. G. and Raikar, Y. N., "Authentication of H.264 streams by direct watermarking of CAVLC blocks," in [*Security, Steganography, and Watermarking of Multimedia Contents IX, Part of IS&T/SPIE 19th Annual Symposium on Electronic Imaging, SSWMC2007, SPIE2007*], **6505** (Feb. 2007).
6. Cox, I. J., Kilian, J., Leighton, T., and Shamoon, T., "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing* **6**(12), 1673–1687 (1997).
7. Golikeri, A., Nasiopoulos, P., and Wang, Z. J., "Robust digital video watermarking scheme for H.264 advanced video coding standard," *Journal of Electronic Imaging* **16**(4) (2007).
8. Shahid, Z., Chaumont, M., and Puech, W., "Fast Protection of H.264/AVC by Selective Encryption of CABAC For I & P Frames," in [*The 17th European Signal Processing Conference, EUSIPCO'2009*], (Aug. 2009).
9. Zou, D. and Bloom, J. A., "H.264/AVC substitution watermarking: a CAVLC example," in [*Media Forensics and Security I, Part of IS&T/SPIE 21th Annual Symposium on Electronic Imaging, MFS2009, SPIE2009*], 7254 (Jan. 2009).
10. Miller, M. L., Doërr, G., and Cox, I. J., "Applying Informed Coding and Informed Embedding to Design a Robust, High Capacity Watermark," *IEEE Transactions on Image Processing* **13**(6), 792–807 (2004).
11. Noorkami, M. and Mersereau, R. M., "Digital Video Watermarking in P-Frames With Controlled Video Bit-Rate Increase," *IEEE Transactions on Information Forensics and Security, TIFS'2008* **3**, 441–455 (Sept. 2008).
12. Li, Q. and Cox, I. J., "Using Perceptual Models to Improve Fidelity and Provide Resistance to Valumetric Scaling for Quantization Index Modulation Watermarking," *IEEE Transactions on Information Forensics and Security* **2**(2), 127–139 (2007).
13. Zhang, J., Li, J.-g., and Zhang, L., "Video watermark technique in motion vector," in [*Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing*], *SIBGRAPI'2001*, 179–182, IEEE Computer Society, Washington, DC, USA (2001).
14. Linnartz, J. P. M. G. and Talstra, J. C., "MPEG PTY-Marks: Cheap Detection of embedded Copyright Data in DVD-Video," in [*Proceedings 5th European Symposium on Research in Computer Security, ESORICS'1998*], **1485**, 221–240 (Sept. 1998).

15. Hartung, F. and Girod, B., "Watermarking of uncompressed and compressed video," *Signal Processing* **66**, 283–301 (May 1998).
16. Haitsma, L. and Kalker, T., "A Watermarking Scheme for Digital Cinema," in [*IEEE International Conference on Image Processing, ICIP'2001*], **1**, 587–489 (Oct. 2001).
17. Gong, X. and Lu, H.-M., "Towards fast and robust watermarking scheme for h.264 video," in [*Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia*], *ISM '08*, 649–653, IEEE Computer Society, Washington, DC, USA (2008).
18. Deguillaume, F., Csurka, G., O'Ruanaidh, J., and Pun, T., "Robust 3D DFT Video Watermarking," in [*Security and Watermarking of Multimedia Contents, Part of IS&T/SPIE 12th Annual Symposium on Electronic Imaging, SPIE'1999*], **3657**, 113–124 (Jan. 1999).
19. Xie, F., Furon, T., and Fontaine, C., "On-off keying modulation and tardos fingerprinting," in [*10th ACM Multimedia and Security Workshop, MM&Sec'2008*], 101–106 (Sept. 2008).
20. Chen, C., Ni, J., and Huang, J., "Temporal Statistic Based Video Watermarking Scheme Robust against Geometric Attacks and Frame Dropping," in [*8th International Workshop on Digital Watermarking, IWDW'2009*], **Guildford, UK**, 81–95 (Aug. 2009).
21. Shahid, Z., Chaumont, M., and Puech, W., "Spread Spectrum-Based Watermarking for Tardos Code-Based Fingerprinting of H.264/AVC Video," in [*IEEE International Conference on Image Processing, ICIP'2010*], (sep. 2010).
22. Kerckhoffs, A., "La Cryptographie Militaire," *Journal des Sciences Militaires* **IX** (pp. 5-38 Jan. 1883, pp. 161-191, Feb. 1883).
23. Cayre, F., Fontaine, C., and Furon, T., "Watermarking Security: Theory and Practice," *IEEE Transactions on Signal Processing* **53**(10), 3976–3987 (2005). special issue "Supplement on Secure Media III".
24. Pérez-Freire, L., Comesana, P., Troncoso-Pastoriza, J. R., and Pérez-González, F., "Watermarking Security: a Survey," *IEEE Transactions on Data Hiding and Multimedia Security* **1**(4300), 41–72 (2006).
25. Doërr, G. and Dugelay, J.-L., "A guide tour of video watermarking," *Signal Processing: Image Communication* **18**(4), 263 – 282 (2003).
26. Mathon, B., Bas, P., Cayre, F., and Macq, B., "Comparison of secure spread-spectrum modulations applied to still image watermarking," *Annals of Telecommunications* **64**, 801–813 (2009).
27. Bas, P. and Westfeld, A., "Two Key Estimation Techniques for the Broken-Arrows Watermarking Scheme," in [*11th ACM workshop on Multimedia and Security, MM&Sec'2009*], 1–8 (Sept. 2009).
28. Xie, F., Furon, T., and Fontaine, C., "Better security levels for 'Broken Arrows'," in [*IS&T/SPIE 22th Annual Symposium on Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents XII*], **7541** (Jan. 2010).
29. Bas, P. and Doërr, G., "Evaluation of an Optimal Watermark Tampering Attack Against Dirty Paper Trellis Schemes," in [*10th ACM workshop on Multimedia and Security, MM&Sec'2008*], 227–232 (Sept. 2008).
30. Pérez-Freire, L. and Pérez-González, F., "Exploiting security holes in lattice data hiding," in [*Information Hiding, IH'2007, Lecture Notes in Computer Science*], 159–173, Springer-Verlag (2007).
31. Tardos, G., "Optimal Probabilistic Fingerprint Codes," in [*ACM symposium on Theory of computing*], 116–125 (2003).
32. Škorić, B., Vladimirova, T. U., Celik, M. U., and Talstra, J., "Tardos fingerprinting is better than we thought," *IEEE Transactions on Information Theory, TIT'2008* **54**(8), 3663 – 3676 (2008).
33. Škorić, B., Katzenbeisser, S., and Celik, M. U., "Symmetric tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes and Cryptography* **46**, 137–166 (February 2008).
34. Blayer, O. and Tassa, T., "Improved versions of tardos' fingerprinting scheme," *Designs, Codes and Cryptography* **48**, 79–103 (July 2008).