On-Off Keying Modulation and Tardos Fingerprinting

Fuchun Xie, Teddy Furon, Caroline Fontaine

On-Off Keying Modulation and Tardos Fingerprinting*

Fuchun Xie
INRIA-Bretagne Atlantique
Research Center
Campus de Beaulieu
35042 Rennes, France
fuchun.xie@inria.fr

Teddy Furon
INRIA-Bretagne Atlantique
Research Center
Campus de Beaulieu
35042 Rennes, France
teddy.furon@inria.fr

Caroline Fontaine
CNRS/IRISA and
INRIA-Bretagne Atlantique
Research Center
Campus de Beaulieu
35042 Rennes, France
caroline.fontaine@irisa.fr

ABSTRACT

We consider a particular design of fingerprinting code for multimedia contents, carefully motivated by a detailed analysis. This design is based on a two-layer approach: a probabilistic fingerprinting code *a la* Tardos coupled with a zero-bit side informed watermarking technique. The detection of multiple watermark presences in content blocks give birth to extended accusation processes, whose performances, assessed experimentally, are excellent. This prevents the colluders from mixing different content blocks, a class of collusion which is not encompassed in the classical marking assumption. Therefore, the collusion must stick to the block exchange strategy which is fully tackled by the fingerprinting code.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image processing and computer vision, Applications

General Terms

Design, Experimentation

Keywords

Watermarking, fingerprinting, anti-collusion

1. INTRODUCTION

This article deals with active fingerprinting, also known as traitor tracing, or forensics, when applied on multimedia content. Fingerprinting is the application where a content server distributes personal copies of the same content to n different buyers. Some are dishonest users, called colluders, who mix their copies to yield a pirated content. This is the so-called collusion process. By analyzing this pirated content, the accusation process (or the decoding) aims at tracing back the colluders' identity. One hot issue in this application is to find the right association of two pieces of technology: an anti-collusion or fingerprinting code and a watermarking technique.

A fingerprinting code is a set of n different m symbol sequences $\{\mathbf{X}_j\}_{j=1}^{n}$. The symbols belong to a q-ary discrete alphabet: $X_j(i) \in \mathcal{X}$, $\forall (j,i) \in [n] \times [m]$, with $|\mathcal{X}| = q$ ($[n]$ denotes $\{1,\ldots,n\}$). The code has the property that observing a mixture of a bounded number of code sequences, the decoding can retrieve a subset of the original sequences used for this forgery.

Each sequence identifying a user has to be hidden in his personal copy with a watermarking technique. The embedding is block based: it divides the content into consecutive blocks and it hides a symbol per block. We assume here that the content (a video or an audio clip) is long enough so that there is at least m blocks. This two-layer approach has two advantages. The blocks of content are watermarked offline in q versions containing a different symbol. The online content server is just a switch that ships the right blocks according to the user sequence. On the other hand, the pirated copy is processed only once by the computationally greedy watermark decoding for retrieving a m symbol sequence $\mathbf{Y}$. Then, the lighter accusation process of the fingerprinting code accuses some users (or nobody) based on this 'pirated' sequence $\mathbf{Y}$.

So far, the designs of these two technologies have often been made separately. The fingerprinting codes have been mostly proposed by the cryptographic community with models of the collusion process defined on the sequence space since the pioneering work [1]. Watermarking techniques are mainly studied by people in the image or signal processing community. Hence, it is crucial to verify that a collusion of watermarked contents is compliant with the assumptions made by fingerprinting designers. Sec. 2 details the attack model on the content space and it shows that its impacts is quite involved for the above layer. There is a class of attack, denoted in the sequel fusion, which can have dramatic effect. However, Sec. 3 proposes an interesting example of this layered approach: a Tardos fingerprinting code [8] with a zero-bit side informed watermarking technique [3] used with a on-off modulation. This constitutes a good counter-attack to the fusion. The experimental investigations of Sec. 4 show that the colluders have no longer interest of using this class

*This work is supported by the French national programme "Securité ET INformatique" under project NEBBIANO, ANR-06-SETIN-009; and this work is also supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT, and by the French ANR/RIAM programme under Contract ESTIVALE

MM&Sec'08, September 22–23, 2008, Oxford, United Kingdom.

of attack, and should restrict their collusion process to collusion modeled by the marking assumption and correctly handled by the fingerprinting code.

2. MODELS OF COLLUSION

2.1 Block based embedding

We suppose the watermarking process starts extracting a long sequence $\mathbf{s}^{(o)}$ of L extracted features (such as DCT, DFT, DWT coefficients...) from the original content. This sequence is split into m blocks of l samples $\{\mathbf{b}_i^{(o)}\}_{i=1}^m$ (we suppose that $L = ml$), s. t. $\mathbf{b}_i^{(o)} = (s^{(o)}(il+1), \ldots, s^{(o)}((i+1)l))$. The watermark embedding hides the symbol $X_j(i)$ into the block $\mathbf{b}_i^{(o)}$ producing the i-th watermarked block $\mathbf{b}_{i,j}^{(w)}$ delivered to the j-th user. Packed back all together, this yields the watermarked sequence $\mathbf{s}_j^{(w)}$.

2.2 Three classes of attacks

A set of colluders $\mathcal{C}$ with the number of members up to c receives their personalized copies and mixes them to forge a pirate copy $\mathbf{s}^{(p)}$. We assume that the collusion process can be expressed as a sample-wise linear transform plus some noise:

$$\forall k \in [L], \quad s^{(p)}(k) = \sum_{j \in \mathcal{C}} w_j(k) s_j^{(w)}(k) + n(k). \qquad (1)$$

Weights are such that $\sum_{j\in\mathcal{C}} w_j(k) = 1$, $\forall k \in [L]$, in order to reconstruct a pirated copy similar to the original sequence. We can also assume that $\sum_{k\in[L]} |w_j(k)| \approx Lc^{-1}$, $\forall j \in \mathcal{C}$ if the colluders participate evenly in the collusion. This general model allows us to make the three following classes.

2.2.1 Block exchange

We refuse to assume that the splitting of the sequence into blocks is a secret primitive and we suppose that there is no way of keeping this process secret. Therefore, the colluders know the blocks. One strategy is to forge the pirated copy by copy-pasting blocks from the colluders' copies. It means that $\forall k \in \{il+1, \cdots, (i+1)l\}$, $w_j(k) = \delta_{J(i)}(j)$, where δ is the Kronecker function and function $J : [m] \mapsto \mathcal{C}$ maps a block index to a colluder index. This class can be divided into families depending on the nature of function J:

- This mapping is independent of any auxiliary data, like, for instance, a random drawing uniformly over $\mathcal{C}$.
- This mapping depends on colluders' blocks. Comparing their blocks for a given index, they can know how many different symbols are embedded in their blocks, and their frequency: for instance, a majority (resp. minority) vote, where the block put in the pirated copy is the most (resp. less) frequent.
- This mapping depends on colluders' symbols. This can be, for instance, a constant symbol strategy, where, whenever possible, the colluders always select the block with a given symbol inside.

These families of attacks are managed by the anti-collusion code because it exactly matches the scenario envisaged by the cryptographers (the so-called marking assumption [1]). However, note that the third subfamily is not relevant a priori in multimedia fingerprinting. Fingerprinting codes invented by cryptographers foresee this case because the content is modelized as a long string of symbols[1] directly observable by the colluders. In multimedia scenario, the colluders do not know the watermarking secret key to decode symbols embedded in their copies. With this respect, these fingerprinting codes are more powerful than we need, however they do not foresee the next class of collusion attacks.

2.2.2 Fusion

The first class has the particularity that the weights are exclusive (one weight equals one and the others zero) and constant over a block. This second class can thus be divided into two following families:

- The weights are non exclusive. Eq. (1) really mixes several samples into one value, possibly with negative weights. This can be, for instance, an average where all the weights equal c^{-1}.
- The weights evolve at a finer granularity than the one of the blocks defined at the watermark embedding. For instance, Eq. (1) selects a sample according to a rank (median, maximum, minimum) among the collection [5].

These weights can also be described as random variables. For instance, the collusion attack enforcing $\text{Prob}(w_j(k) = 1) = 1/2$ for indices such that

$$j \in \{\arg\max_{j\in\mathcal{C}}\{s_j^{(w)}(k)\}, \arg\min_{j\in\mathcal{C}}\{s_j^{(w)}(k)\}\},$$

is known to yield a low SNR at the decoding side of the watermark [9].

2.2.3 Content processing

The last class of attack corresponds to regular content processing, such as lossy source coding, lowpass filtering, denoising, which can remove the presence of the watermarking signal. This is encompassed in our model by the addition of a noise $\mathbf{n}$ in Eq. (1). This class of attack can be used by a dishonest user alone, or by a group of colluders in addition to the first or second class.

2.3 When can we trace colluders?

The second and third classes of attacks are not considered by typical cryptographic fingerprinting codes. It is up to the watermarking layer to tackle these classes. Of course, it will successfully do so if the decoded symbol $Y(i)$ belongs to the set $\{X_j(i)\}_{j\in\mathcal{C}}$ as assumed by the marking assumption.

The robustness of the watermarking technique is of the utmost importance to fight against the third class. Yet, some fingerprinting codes handle symbol erasures. In the same way, the impact of the mixing of several watermarked blocks on the decoded symbol is an even more involved problem. Some fingerprinting code still work if, for some indices, the decoded symbol is not in the subset $\{X_j(i)\}_{j\in\mathcal{C}}$. However, these two watermarking decoding failures must seldom occur as the cost to be paid is much longer code sequences.

[1]Some of them being substituted by the symbols of the codeword.

3. ON-OFF KEYING MODULATION

3.1 Positive rate watermarking

Let us write that the watermarked signal is the sum of the original and watermark signal: $\mathbf{b}_{i,j}^{(w)} = \mathbf{b}_i^{(o)} + \mathbf{w}(X_j(i), \mathbf{b}_i^{(o)})$. Consider a fusion of c signals via an average process. Then, the pirated block reads $\mathbf{b}_i^{(p)} = \mathbf{b}_i^{(o)} + c^{-1}\mathbf{w}(X_j(i), \mathbf{b}_i^{(o)}) + \boldsymbol{\epsilon}$. This is very different from attacks of the class 2.2.3 because of the scaling factor c^{-1} and the noise $\boldsymbol{\epsilon}$, sum of the other watermark signals, which is not at all independent of the host or the watermark signals. It is not sure that a very robust watermarking technique greatly performing against the third class, is actually good against the fusion of blocks.

From a geometrical point of view, the watermark decoding output a symbol whenever the input block belongs to its decoding region. There are two possibilities: either the space is a partition of q decoding regions, either it is a partition into these q regions plus one for the erasure. Assume that the embedding algorithm succeeded to push the host signal in the different decoding region, then the average attack amounts to take the barycentre of points of these regions. There are three possibilities:

- The barycentre still belongs to the decoding region associated to one of the colluders' symbols,
- The barycentre is inside another decoding region,
- The barycentre is inside the erasure region.

As already mentioned, fingerprinting codes 'dislike' the two last cases, some can manage them if and only if this occurs very rarely and at the price of longer code sequences. Knowing this, the colluders will prefer this class of attacks.

3.2 Zero-bit watermarking

Zero-bit watermarking is similar to on-off keying (OOK) in digital communications. This modulation is used on very rare applications: fiber communication where it is not possible to modulate the light emission, except by switching it on and off. Some theoretical works also show that OOK is the last solution to communicate when the channel transmission quality is really too bad (e.g., the delay spread of the fading is less than the symbol duration, so that channel estimation and equalization is not possible) [4]. The use of zero-bit watermarking is not new in multimedia fingerprinting. For instance, Safavi-Naini and Yang embed q-ary symbols in pictures using q different secret keys of a classical spread spectrum scheme [6]. We use a different zero-bit watermarking technique which is side-informed.

The q possible watermark signals $\{\mathbf{w}(X, \mathbf{b}_i^{(o)})\}_{X\in\mathcal{X}}$ are not strictly independent because all taking advantage of the side-information $\mathbf{b}_i^{(o)}$. But they are less dependent compared to signals from a positive rate watermarking technique, because they are generated from q independent secret keys. Hence, a fusion attacks is more similar to the scaling and the addition of an independent noise. Another advantage is that it is very unlikely that the barycenter is inside the detection region of symbol (i.e., a secret key) which doesn't belong to $\{X_j(i)\}_{j\in\mathcal{C}}$. The rationale is that, for a very small probability of false alarm, the colluders cannot succeed to watermark a block without knowing this secret key from signals which are independent from this detection region. Another way to see this, is that, assuming the fusion is linear, the forged block remains in an affine space passing by the point $\mathbf{b}_i^{(o)}$ and spanned by the watermark signals $\{\mathbf{w}(X_j(i), \mathbf{b}_i^{(o)})\}_{j\in\mathcal{C}}$, which is almost orthogonal to the detection region related to the other keys. Hence, this event should be as rare as a false alarm.

3.3 Past approaches

So far, we have defended the fact that a zero-bit watermarking scheme can avoid the second unwanted possibility of the fusion attack. Furthermore, as the zero-bit watermaking is also more robust than positive rate watermarking, the third unwanted possibility is also less likely. We would like now to stress another advantage. The number of detection outputs is indeed $2^q > q$, as, for each of the q secret keys, the detector will give a binary decision. Hence there are cases where several watermark signals are detected. At block i, a set of symbols $\mathcal{Y}_i = \{Y_i(k)\}_{k=1}^{K_i}$ is detected. K_i represent the number of symbols detected at block i. What kind of fingerprinting code can take advantage of this feature? We found in literature the following two candidates.

Many strong c-traceable code are based on algebraic error correcting codes such as Reed-Solomon codes. This feature allows two strategies: list decoding or iterative decoding. List decoding finds a group of nearest code sequences (from the pirated sequence) [7] beyond the decoding distance, and its algorithm like Guruswami-Sudan takes into account some reliability measures about the decoded symbols, which could be based on the decoded symbols $\mathbf{Y}_i$. Another strategy is to decode iteratively the pirated sequence to find several colluders. In [2], symbols of the pirate sequence are replaced by erasures when they match symbols of code sequences decoded in previous iterations. This new pirated sequence is again decoded at the next iteration. Here, we can replace erasure by another symbol decoded in the block.

3.4 Our approach

A well known weak traceable code is the probabilistic Tardos fingerprinting code, and especially its q-ary version proposed by Skoric *et al.* [8]. $\{\mathbf{p}_i\}_{i=1}^m$ are auxiliary vectors used for generating the code: Symbols $X_j(i)$ are independent random variables drawn such that $\text{Prob}(X_j(i) = X) = p_i(X)$, for $X \in \mathcal{X}$. Thus, we have $\mathbf{p}_i^T\mathbf{1} = 1$. Skoric *et al.* propose to draw each $\mathbf{p}_i$ independently from a Dirichlet distribution with shape parameter κ. The accusation process first calculates a score S_j for the j-th user:

$$S_j = \sum_{i=1}^{m} U(Y(i), X_j(i), \mathbf{p}_i). \tag{2}$$

A focused decoding accuses user j if $S_j > Z$, a general decoding accuses users with the biggest scores. Skoric *et al.* use the same summands as Tardos:

$$U(Y, X, \mathbf{p}) = \delta_Y(X)g_1(p(Y)) + (1 - \delta_Y(X))g_0(p(Y)), \tag{3}$$

with $g_1(p) = \sqrt{(1-p)/p}$ and $g_0 = -\sqrt{p/(1-p)}$. Our work is to extend this decoding in order to take into account the fact that a list of symbols, denoted $\mathcal{Y}_i = \{Y_i(1), \ldots, Y_i(K_i)\}$, are decoded from the i-th block. The i-th watermark detection doesn't bring any information about the guilt of user j when the list is empty ($K_i = 0$ and $\mathcal{Y}_i$ is an empty set) or full ($K_i = q$ and $\mathcal{Y}_i = \mathcal{X}$) since all users have then a decoded symbol.

3.4.1 First method

We propose the following score, with U defined in (3):

$$S_j = \sum_{i=1}^{m} \sum_{k=1}^{K_i} U(Y_i(k), X_j(i), \mathbf{p}_i). \quad (4)$$

At first sight, it is as if the code length would have increased from m to $m\bar{K}$, with $\bar{K} = m^{-1} \sum_{i=1}^{m} K_i$. The longer a Skoric's code is, the more reliable is the accusation process. This rationale justifying the idea isn't correct because the summands are not independent. The experimental section shows however that it works great.

3.4.2 Second method

The sum is defined in (2), where the summands are:

$$U(\mathcal{Y}, X, \mathbf{p}) = \delta_{\mathcal{Y}}(X) g_1(p_{\mathcal{Y}}) + (1 - \delta_{\mathcal{Y}}(X)) g_0(p_{\mathcal{Y}}), \quad (5)$$

with $\delta_{\mathcal{Y}}(X) = 1$ if $X \in \mathcal{Y}$, else 0, and $p_{\mathcal{Y}} = \sum_{k=1}^{K} p(Y(k))$. Our rationale here is to decrease the variance of the colluders' scores: whatever their symbol $X_j(i) \in \mathcal{Y}_i$, they receive the same penalization $g_1(p_{\mathcal{Y}_i})$.

4. EXPERIMENTAL WORKS

The first experimental work evaluates the performances of the watermarking technique in order to tune accordingly the fingerprinting code.

4.1 Evaluation of the watermarking technique

The zero-bit watermarking technique 'Broken Arrows' [3] is used as a practical watermarking solution in our experimentation. Its performances in terms of robustness, security, and imperceptibility are state-of-the-art. Its detector runs very fast thanks to a simple and efficient implementation. Some modifications of the code further improve detection speed. After a wavelet transform, N_v correlations onto secret carriers are calculated. This vector is divided into q sets of $N_c = N_v/q$ components each. In the original algorithm, the watermark embedding uses the most host-correlated direction of the first set as a secret vector $\mathbf{v}'_C$ (see [3, Eq.(18)]). In the same way, the detection looks whether the received vector is inside one of the N_c hypercones defined by the directions of the first set (see [3, Eq.(5)]). Here, everything remains the same except that the set in use at the embedding is given by the symbol $X_j(i)$. These sets of secret directions are independent, whence all is as if the embedding was done with q different secret keys. The detection outputs the indices of the sets which have given a positive output (the signal is inside one of their hypercones).

We have used 2000 images to evaluate the collusion resistance performance of this watermarking solution. The PSNR of the watermarked images is around 43 dB. When an average collusion attack is applied from ℓ different fingerprinted images followed by a JPEG compression with a quality factor $Q = 20$, we compute the probability $P(K|\ell)$ of detecting K different watermarks. Table 1 shows the result. The detection probability of the proposed watermarking system is quite good (~ 0.95) when ℓ is small (1 or 2). The performance becomes worse when mixing more than 2 watermarked images because the strength of one watermark becomes smaller as more images are averaged. For $\ell = 4$, half of the time, we are not able find any watermark. The maximum number of averaged watermarked images being $\min(q, c)$, there is no point in having q higher than 4.

ℓ	$K=0$	$K=1$	$K=2$	$K=3$	$K=4$
1	0	1	0	0	0
2	0.01	0.03	0.96	0	0
3	0.17	0.23	0.26	0.34	0
4	0.46	0.23	0.15	0.11	0.05

Table 1: The conditional probabilities $P(K|\ell)$ of detecting K watermarks when the average of ℓ watermarked images is JPEG compressed with a quality factor $Q = 20$.

4.2 Evaluation of the fingerprinting code

The methods presented in Secs. 3.4.1 and 3.4.2 amount to the same accusation process as Skoric's when $\mathcal{Y}$ is a singleton. This occurs when the colluders choose the block exchange class of Sec. 2.2.1. According to Skoric *et al.*, one of their best attacks within this class is the so-called 'extremal' strategy defined in [8, Eq.(58)]. These authors also noticed that there exist an optimal shape parameter κ to counter-attack this worst case scenario.

Fig. 1 (resp. 2) shows the experimental measures of the expectations (resp. the variances) of the scores of an innocent $\mu_{I,0}$ and of a colluder $\mu_{C,0}$ (resp. $\sigma^2_{I,0}$ and $\sigma^2_{C,0}$). We set $m = 300$, $q = 4$, $c = 20$ and κ is varying from 0.1 to 0.5. Noticeable features of Skoric detection are that $\mu_{I,0} = 0$ and $\sigma^2_{I,0} = m$. The Kullback Leibler distance between the two pdfs, assuming that the scores are Gaussian distributed, roughly show the performances of the focused accusation process: the higher D_{KL} is, the more powerful is the test.

$$D_{KL}(I; C) = \frac{1}{2} \left(\frac{(\mu_I - \mu_C)^2}{\sigma^2_C} + \frac{\sigma^2_I}{\sigma^2_C} - 1 + \log \frac{\sigma^2_C}{\sigma^2_I} \right). \quad (6)$$

Fig. 3 shows that $\kappa = 0.23$ is optimal for this experimental setup, which more or less confirms Skoric *et al.* optimal value of 0.27. The slight difference is not surprising because we use a completely different optimality criterion.

4.3 Evaluation of the new methods

The new methods enter in the picture when the collusion chooses the fusion strategy of Sec. 2.2.2. We repeat that classical cryptographic fingerprinting codes are not designed for this kind of collusion. Our proposal (q-ary Tardos code, zero-bit watermarking, and improved accusation sums) raises interests if we can show that the fusion strategy is worse from the colluders' point of view. Therefore, the collusion will reject it and it will stick to the block exchange strategy, for which fingerprinting codes have been designed.

We first investigate how frequently our methods yield different score than the regular Skoric's accusation process. One necessary condition is that c colluders have more than one hidden symbol at the i-th block. Table 2 shows that this occurs with a probability greater than 0.57 if $c \geq 3$. Another condition is that the number of decoded symbols after the fusion is neither 0 nor q, else the summands at that index are zeros. The performance of the watermarking technique against the fusion has a clear impact on this condition. Combining tables 1 and 2, we easily have the probability of decoding K symbols: $P(K|c) = \sum_{k=1}^{\min(q,c)} P(K|\ell) P(\ell|c)$. Table 3 shows that $P(K = q|c)$ is negligible, and that $P(K = 0|c)$ is slowly increasing with c thanks to the good robustness

c	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
2	0.60	0.40	0	0
3	0.43	0.50	0.07	0
4	0.34	0.52	0.14	0.00
5	0.28	0.52	0.19	0.01
6	0.24	0.51	0.23	0.02
10	0.15	0.46	0.33	0.06
15	0.11	0.40	0.38	0.10
20	0.08	0.36	0.42	0.14

Table 2: The conditional probabilities $P(\ell|c)$ that the colluders have ℓ watermarked versions of a block for $2 \leq c \leq 20$ and $\kappa = 0.23$.

c	$K = 0$	$K = 1$	$K = 2$	$K = 3$	$K = 4$
2	0.00	0.61	0.39	0	0
3	0.02	0.46	0.49	0.03	0
4	0.03	0.39	0.53	0.05	0.00
5	0.04	0.34	0.55	0.07	0.00
6	0.05	0.32	0.55	0.08	0.00
10	0.09	0.26	0.53	0.12	0.00
15	0.12	0.23	0.50	0.14	0.01
20	0.14	0.22	0.47	0.16	0.01

Table 3: The probabilities $P(K|c)$ of detecting K watermarks per block when the collusion size is c.

of the zero-bit watermarking technique. Even for $c = 20$, our methods are active over 63% blocks.

The experimentation setup is the same as described in Sec. 4.2. The collusion is based on Table 1 to simulate a fusion: whenever the collusion has ℓ different symbols, we randomly pick up K of them. Statistics are established from 32,000 scores for the innocents and 8,000 scores for the colluders. Fig. 1 shows that the expectation of an innocent's score is zero whereas the one of the colluder is roughly the same for both methods and especially much higher than previously. Fig. 2 shows that the variance of the scores (innocent's and colluder's) are smaller than previously for both methods. The first method is very good at lowering σ_I^2 whereas the second method has the smallest σ_C^2. The overall performance measured by the Kullback Leibler distance confirms in Fig. 3 that the collusion has no interest in adopting the fusion strategy.

A practical issue is the value of the threshold Z. The following relationship holds for both methods:

$$\mu_I = \mu_{I,0}\,, \quad \mu_C \geq \mu_{C,0} \tag{7}$$

$$\sigma_I^2 \leq \sigma_{I,0}^2\,, \quad \sigma_C^2 \leq \sigma_{C,0}^2 \tag{8}$$

Therefore, if the length of the code is large enough to ensure required probabilities of false alarm and false negative when comparing the scores to Z for the block exchange class of attack, then, this threshold will ensure even lower probabilities of errors for the fusion class thanks to the performance of our methods. This statement is true only when the Gaussian assumption holds, *i.e.* for m large enough.

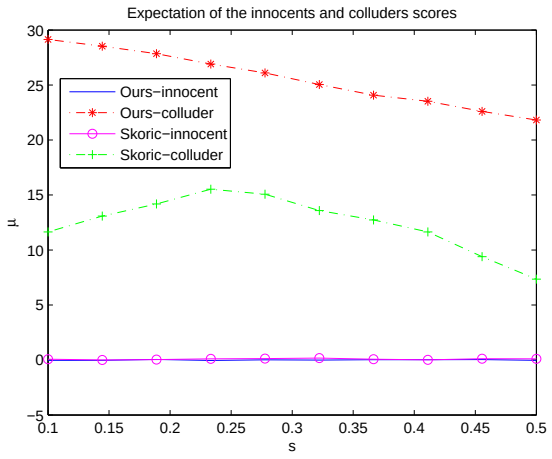

Figure 1: Expectation of an innocent's (solid) and a colluder's (dash) score against Dirichlet distribution shape parameter κ for block exchange class (green), fusion class and first method (blue), fusion class and second method (red).

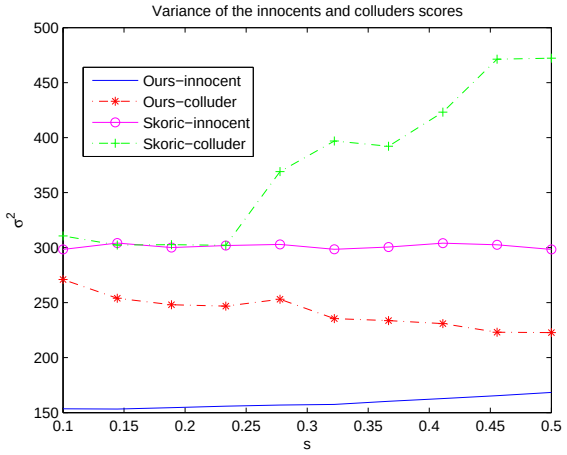

Figure 2: Variance of an innocent (solid) and a colluder's (dash) score against Dirichlet distribution shape parameter κ for block exchange class (green), fusion class and first method (blue), fusion class and second method (red).

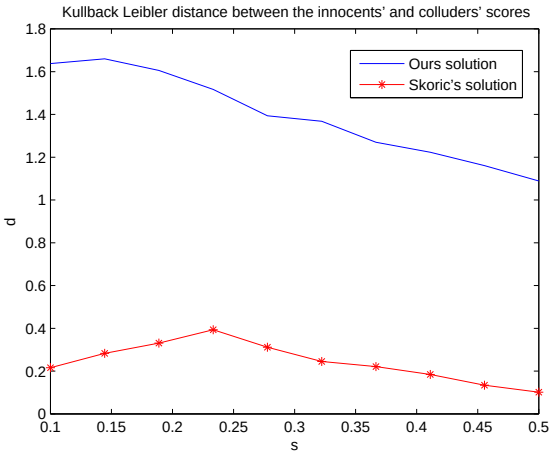

Figure 3: Kullback Leibler distance between the innocent's and colluder's scores pdf against Dirichlet distribution shape parameter κ for block exchange class (green), fusion class and first method (blue), fusion class and second method (red).

5. CONCLUSION

The proposed design has three ingredients: a symmetric q-ary Tardos fingerprinting code, a state of the art zero-bit side informed watermarking technique used with a on-off keying modulation, and an extended accusation process taking into account list of decoded symbols. Our experimental study shows that these ingredients blend into a very good design because it completely shuts down the fusion class of attacks. Following this strategy, the collusion helps more the accusation process than it deludes it. The collusion is then back to the block exchange class of attack which is fully tackled by the fingerprinting code.

6. REFERENCES

[1] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inform. Theory*, 44:1897–1905, September 1998.

[2] M. Fernandez and M. Soriano. Soft-decoding tracing in fingerprinted multimedia content. *IEEE Multimedia*, 11(2):38–46, 2004.

[3] T. Furon and P. Bas. Broken arrows. *submitted to EURASIP Jounal on Information Security*, 2008.

[4] M. Gursoy, H. Poor, and S. Verdú. On-off frequency-shift keying for wideband fading channels. *EURASIP Journal on wireless communications and networking*, 2006(ID 98564):15 pages, 2006.

[5] P. Moulin and N. Kiyavash. Performance of random fingerprinting codes under arbitrary nonlinear attacks. In *Proc. ICASSP*, Honolulu, avril 2007.

[6] R. Safavi-Naini and Y. Wang. Collusion-secure q-ary fingerprinting for perceptual content. In Springer-Verlag, editor, *Proc. Security and Privacy in Digital Rights Management, SPDRM'01*, volume 2320 of *Lecture Notes in Computer Science*, pages 57–75, 2001.

[7] A. Silverberg, J. R. Staddon, and J. Walker. Application of list decoding to tracing traitors. *IEEE Trans. Inform. Theory*, 49:1312–1318, may 2003.

[8] B. Skoric, S. Katzenbeisser, and M. Celik. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. *Designs, Codes and Cryptography*, 46(2):137–166, February 2008.

[9] Z. Wang, M. Wu, H. Zhao, W. Trappe, and K. Liu. Resistance of orthogonal gaussian fingerprints to collusion attacks. In *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, pages 724–727, Hong Kong, April 2003. IEEE ICASSP'03.

APPENDIX

A. FIRST METHOD: $\mu_I = 0$

We have $\mu_I = m\mathbb{E}(\sum_{Y\in\mathcal{Y}} U(Y,X,\mathbf{p}))$ giving:

$$\begin{aligned}
&\mu_I m^{-1}\\
&= \sum_{\mathcal{Y},X} p_X p_{\mathcal{Y}} \sum_{Y\in\mathcal{Y}} \delta_Y(X) g_1(p_Y)) + (1-\delta_Y(X)) g_0(p_Y)\\
&= \sum_{\mathcal{Y}} p_{\mathcal{Y}} \sum_{X,Y\in\mathcal{Y}} p_X(\delta_Y(X) g_1(p_Y) + (1-\delta_Y(X)) g_0(p_Y))\\
&= \sum_{\mathcal{Y}} p_{\mathcal{Y}} \sum_{Y\in\mathcal{Y}} \sqrt{p_Y(1-p_Y)} - \sqrt{p_Y(1-p_Y)}\\
&= 0
\end{aligned}$$

B. SECOND METHOD: $\mu_I = 0$

We have $\mu_I = m\mathbb{E}(U(\mathcal{Y},X,\mathbf{p}))$ giving:

$$\begin{aligned}
&\mu_I m^{-1}\\
&= \sum_{\mathcal{Y},X} p_X p_{\mathcal{Y}} (\delta_{\mathcal{Y}}(X) g_1(p_{\mathcal{Y}})) + (1-\delta_{\mathcal{Y}}(X)) g_0(p_{\mathcal{Y}}))\\
&= \sum_{\mathcal{Y}} p_{\mathcal{Y}} \sum_{X} p_X(\delta_{\mathcal{Y}}(X) g_1(p_{\mathcal{Y}}) + (1-\delta_{\mathcal{Y}}(X)) g_0(p_{\mathcal{Y}}))\\
&= \sum_{\mathcal{Y}} p_{\mathcal{Y}} (\sqrt{p_{\mathcal{Y}}(1-p_{\mathcal{Y}})} - \sqrt{p_{\mathcal{Y}}(1-p_{\mathcal{Y}})})\\
&= 0
\end{aligned}$$

C. SECOND METHOD: σ_I^2

We have $\sigma_I^2 = m\mathbb{E}(U(\mathcal{Y},X,\mathbf{p})^2)$ giving:

$$\begin{aligned}
&\sigma_I^2 m^{-1}\\
&= \sum_{\mathcal{Y}\notin\{\emptyset,\mathcal{X}\},X} p_X p_{\mathcal{Y}} (\delta_{\mathcal{Y}}(X) g_1(p_{\mathcal{Y}}))\\
&+ (1-\delta_{\mathcal{Y}}(X)) g_0(p_{\mathcal{Y}}))^2\\
&= \sum_{\mathcal{Y}\notin\{\emptyset,\mathcal{X}\}} p_{\mathcal{Y}} (\sum_{X\in\mathcal{Y}} p_X g_1(p_{\mathcal{Y}})^2 + \sum_{X\notin\mathcal{Y}} p_X g_0(p_{\mathcal{Y}})^2)\\
&= \sum_{\mathcal{Y}\notin\{\emptyset,\mathcal{X}\}} p_{\mathcal{Y}}(1-p_{\mathcal{Y}}+p_{\mathcal{Y}}) = 1 - p_{\mathcal{Y}}(\emptyset) - p_{\mathcal{Y}}(\mathcal{X})
\end{aligned}$$