Expectation Maximization decoding of Tardos probabilistic fingerprinting code

Ana Charpentier[1], Fuchun Xie[1], Caroline Fontaine[2], and Teddy Furon[3]

[1] IRISA/INRIA-Rennes research center, Campus de Beaulieu, 35042 Rennes, France
[2] CNRS-IRISA and INRIA-Rennes research center, Campus de Beaulieu, 35042 Rennes, France
[3] Thomson Security Laboratory, 35510 Cesson Sevigne, France

ABSTRACT

This paper presents our recent works on multimedia fingerprinting, improving both the fingerprinting code and the watermarking scheme. Our first contribution focuses on deriving a better accusation process for the well known Tardos codes. It appears that Tardos orginal decoding is very conservative: its performances are guaranteed whatever the collusion strategy. Indeed, major improvements stem from the knowledge of the collusion strategy. Therefore, the first part of this paper investigates how it is possible to learn and adapt to the collusion strategy. Our solution is based on an iterative algorithm *a la* EM, where a better estimation of the collusion strategy yields a better tracing of the colluders, which in return yields a better estimation of the collusion strategy etc.

The second part of this paper focuses on the multimedia watermarking scheme. In a previous paper, we already used the 'Broken Arrows' technique as the watermarking layer for multimedia fingerprinting. However, a recent paper from A. Westfeld disclosed a flaw in this technique. We present here a counter-measure which blocks this security hole while preserving the robustness of the original technique.

Keywords: Watermarking, Proportional embedding, Fingerprinting, Anti-collusion, Tardos code

1. INTRODUCTION

This paper deals with multimedia fingerprinting, also known as transactional watermarking, traitor tracing, copy serialization. The addressed problem is the following: a multimedia content server distributes personal copies of the same content to n different buyers. Some dishonest users, called *colluders* in the sequel, mix their copies to forge a pirated content they will illegally redistribute. An accusation process aims at tracing back the colluders' identity by analyzing this pirated content. The pivotal techniques in this application are a robust watermarking technique, a short anti-collusion fingerprinting code, and their suitable association.

This paper makes an account of the improvements of our works in multimedia fingerprinting. In a first previous paper,[1] we studied the probabilistic fingerprinting code proposed by Tardos. This code has the theoretically minimal order code length.[2] Some enhancements were proposed, provided some assumptions about the colluders hold: their number, the strategy they used to create the pirated copy, ... But these assumptions may not be realistic in real life application. Therefore, in the first part of this paper, we propose a practical way to gain knowledge on the collusion, in order to enforce these enhancements in more realistic conditions.

In a second paper,[3] we studied the association between such a probabilistic fingerprinting code and a zero-bit watermarking technique. To illustrate the advantages of this association, experimental results were based on the 'Broken Arrows' technique used in the BOWS-2 contest (Break Our Watermarking Scheme). However, the lessons learned from this contest showed potential threats under the fingerprinting application scenario. In the second part of this paper, we propose ways to fill the lack of robustness flaw.

Further author information: (Send correspondence to Teddy Furon : E-mail: Teddy.Furon@inria.fr, Telephone: +33 2.99.84.71.98)

2. IMPROVEMENT OF TARDOS FINGERPRINTING CODE: EM DECODING AND ITERATIVE OPTIMIZATION OF THE ACCUSATION FUNCTIONS

2.1 Symmetric Tardos fingerprinting code: limits of previous studies

In 2003, G. Tardos published efficient binary probabilistic fingerprinting codes.[2] They are particularly interesting because of their minimal order length; moreover, they provide a good control of the probabilities of error and are easy to implement. B. Skoric *et al* proposed a symmetric version of these codes, and extended the study to the q-ary case.[4] Both papers used a generic accusation function, the same for any kind of collusion attack. Recently, Furon *et al*[1] showed not only why these accusation functions are optimal in a generic context, but also that more efficient functions can be derived for particular attacks. Here, we propose to go further in this direction, deriving new and more powerful accusation functions.

Before going deeper in our solution, let us briefly present the binary symmetric Tardos codes.[4] We refer the reader to the mentioned papers for more details. Let n denote the number of users, and m the length of the code. The distributed codewords can be arranged as an $n \times m$ matrix $\mathbf{X}$, User j being related to the binary codeword $\mathbf{X}_j = (X_{j1}, X_{j2}, \dots, X_{jm})$. To generate this matrix, m real numbers $p_i \in [0, 1]$ are generated, each of them being randomly drawn according to the probability density function $f(p) = \frac{1}{\pi\sqrt{p(1-p)}}$. We set $\mathbf{p} = (p_1, \dots, p_m)$. Each element of the matrix $\mathbf{X}$ is then independently randomly drawn, following $\mathbb{P}(X_{ji} = 1) = p_i$. The n codewords can then be hidden into multimedia content to identify any of the n users. At the accusation side, we decode the sequence $\mathbf{Y}$ from the pirated copy. To state if User j is involved in the production of this pirated copy, we derive an accusation score S_j. If this score is greater than a given threshold Z, then User j is considered guilty. The scores are computed according to four accusation functions, reflecting the impact of the correlation between the sequence $\mathbf{X}_j$, associated with User j, and the decoded sequence $\mathbf{Y}$:

$$S_j = \sum_{i=1}^{m} U(Y_i, X_{ji}, p_i), \tag{1}$$

with the accusation functions

$$U(1,1,p) = g_{11}(p), \qquad U(0,0,p) = g_{00}(p),$$
$$U(0,1,p) = g_{01}(p), \qquad U(1,0,p) = g_{10}(p).$$

In the usual symmetric codes,[4] these four functions were forced to fulfill some constraints, giving $g_{11}(p) = g_{00}(1-p) = -g_{01}(p) = -g_{10}(1-p) = \sqrt{\frac{1-p}{p}}$.

These codes are really efficient, but their use in practical schemes may lead to constraints that do not necessary coincide with the ones mentioned in the previous papers. This is the reason why we investigate new constraints for their use, and how to adapt them to keep them efficient.

How to choose function f and the accusation functions properly? In his seminal paper,[2] Tardos set the three functions f, g_{10} and g_{11} (called g_0 and g_1; g_{00} and g_{01} were initially equal to 0) and proved that with this choice the scores do not depend on the colluders' strategy. But he did not thoroughly explain how he chose these functions. These choices were kept by Skoric *et al*,[4] who proposed a symmetric scheme, setting g_{00} and g_{01} to non-zero values. Recently, Furon *et al*[1] showed that these functions are optimum with respect to the original assumptions, but can be improved if some of them are relaxed. They showed that the knowledge of the maximum collusion size c_{max} (whatever the collusion strategy) helps us to determine a better function f (impact on the code initialization). They also showed that the knowledge of the collusion strategy helps us to determine better accusation functions (impact on the scores). When the accusation functions match the collusion strategy, the expectation of the scores of the colluders is greater than the expectation for Tardos' scores; but, yet, a mismatch may have a dramatic effect. Hence, they opened a very promising way to greatly improve these codes efficiency. However, their assumption about the collusion strategy was open to criticism: they argued to optimize the accusation functions according to the colluders' strategy, but kept an original constraint (Eq. (14) of[1] – which can be rewritten with our notation as $(1-p)g_{00}(p) = pg_{11}(p)$) – related to the independence between the variance of the innocents scores and the colluders' strategy.

Here we relax this assumption, keeping as few constraints on the accusation functions as possible, and optimizing them according to an estimation of the colluders' strategy. To evaluate the performances of our solution, we compare it with the symmetric Tardos codes of Skoric *et al*[4] and the optimization of Furon *et al*.[1]

How to shorten the code length? One of the more interesting properties of Tardos codes is their minimal length. This length is constrained by the maximum collusion size and the probabilities of errors (accusing an innocent, and missing a colluder). Nevertheless, in real schemes we may have to use shorter codes, as the length is, above all, constrained by the size of the protected content and the embedding rate of the chosen watermarking technique. In this case, it is tempting to use Tardos codes for their easy implementation, and to adapt the accusation process to manage the shorter length and keep a good tracing efficiency.

Here we propose to use EM in an iterative process to get an accurate accusation even if the code length is smaller than the original Tardos' one.

How to estimate the accuracy of the scores? In the original accusation process, we check any User j independently and do not need to compute all the scores to get a verdict. This user is considered as a colluder if his score is greater than the threshold Z. The threshold is chosen to assess a probability of false alarm. Hence, when accusing a user we get an upper bound on the probability of error, but do not have a precise estimation of this error.

Here we adopt another strategy, computing the scores for all the users, and then accusing the user with the highest score. Note that, although this user is the most likely to be guilty, there is no guarantee of not accusing an innocent. Contrary to Tardos, there is no threshold or code length to assess a probability of false alarm. However, some preliminary tests convinced us that it is possible to accuse the biggest score while estimating the probability to be wrong.[5]

2.2 Improving the accusation through an iterative estimation of the colluders' strategy

Following the steps of Furon *et al*,[1] our goal is to optimize the accusation functions according to the colluders' strategy. This is achieved in two stages: first we estimate the strategy, and second we optimize the accusation functions. This process is iterated, each iteration taking advantage of a new estimation of the set of colluders *via* Expectation-Maximization (EM).

We model the collusion strategy by the set of probabilities $\{\mathbb{P}(Y_i = 1|\Sigma_i = \sigma_i), \sigma_i = 0..c\}_{i=1..m}$; the random variable $\Sigma_i = \sum_{j\in\mathcal{C}} X_{ij}$ denotes the number of colluders' symbols equal to 1 at position i. We assume that the same strategy is used for any position $1 \leq i \leq m$. Then, in order to lighten the formulas, we omit the subscript i that indicates the considered position and denote $\boldsymbol{\theta}$ the generic collusion model: $\boldsymbol{\theta} = \{\mathbb{P}(Y = 1|\Sigma = \sigma), \sigma = 0..c\}$. We now describe the steps of the iterative estimation process in details, using the lightened notation.

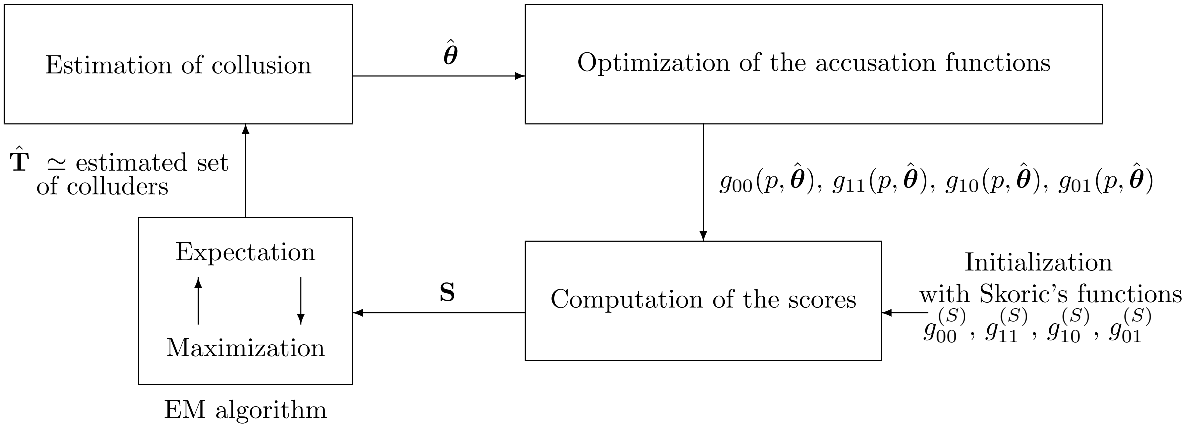
S1. Initialization: We compute all the accusation scores with Skoric *et al*'s accusation functions $g_{11}^{(S)}(p) = g_{00}^{(S)}(1-p) = -g_{01}^{(S)}(p) = -g_{10}^{(S)}(1-p) = \sqrt{\frac{1-p}{p}}$. These scores are stored in the vector $\mathbf{S}$.

S2. EM decoding: The sequence $\mathbf{S}$ amounts to a mixture of scores of innocents and colluders. A classical EM algorithm estimates the status, "innocent" or "colluder", for each user's score. EM takes as inputs the sequence $\mathbf{S}$ and the theoretical means and variances of colluders' and innocents' scores. It outputs the sequence $\hat{\mathbf{T}}$, $\hat{T}_j$ denoting the estimation of the probability that the score S_j of User j is the one of a colluder.

S3. With $\hat{\mathbf{T}}$ and $\mathbf{S}$, we estimate the collusion size $\hat{c}$ and the collusion strategy, namely $\hat{\boldsymbol{\theta}}$.

S4. With respect to $\hat{\boldsymbol{\theta}}$ we optimize the accusation functions $g_{00}(p, \hat{\boldsymbol{\theta}})$, $g_{11}(p, \hat{\boldsymbol{\theta}})$, $g_{10}(p, \hat{\boldsymbol{\theta}})$, and $g_{01}(p, \hat{\boldsymbol{\theta}})$.

S5. We compute the new scores, which are stored in sequence $\mathbf{S}$. We go back to Step S2 to iterate.



Estimation of collusion
$\hat{\boldsymbol{\theta}}$
Optimization of the accusation functions
$\hat{\mathbf{T}} \simeq$ estimated set of colluders
$g_{00}(p,\hat{\boldsymbol{\theta}})$, $g_{11}(p,\hat{\boldsymbol{\theta}})$, $g_{10}(p,\hat{\boldsymbol{\theta}})$, $g_{01}(p,\hat{\boldsymbol{\theta}})$
Expectation
Maximization
EM algorithm
$\mathbf{S}$
Computation of the scores
Initialization with Skoric's functions
$g_{00}^{(S)}$, $g_{11}^{(S)}$, $g_{10}^{(S)}$, $g_{01}^{(S)}$

Figure 1. Computation of new accusation sum functions

Note that the optimized accusation functions derived in[1] ensure that the innocents scores are un-correlated. For a large enough length m, the scores are deemed as Gaussian distributed (sum of i.i.d. random variables). Therefore, the innocent scores are independent. This is a necessary condition for applying the EM algorithm at Step 2. However, this assumption is wrong for colluders' scores. Nevertheless, our experimental tests show that this does not prevent the convergence of this algorithm for a small number of colluders, that is, $c \ll n$.

We provide below more details on the key steps.

2.2.1 Estimation of the strategy : Step S3

We consider the scores $\mathbf{S}$ and the associated probabilities in the vector $\hat{\mathbf{T}}$: $\hat{T}_j = 1$ means User j is a colluder, $\hat{T}_j = 0$ means he is innocent. We estimate the collusion size as $\hat{c} = \lceil \Sigma_{j=0}^{n} \hat{T}_j \rceil$. We only consider the $\hat{c}$ users with the highest probabilities $\hat{T}_j$, and use their sequences to compute the collusion model $\hat{\boldsymbol{\theta}}$. For each position i, we compute σ_i the sum of X_{ji} for all Users j in the estimated colluders set. For each possible value of $0 \leq \sigma_i \leq c$, we compute the mean of the corresponding components of $\mathbf{Y}$.

2.2.2 Optimization of the accusation functions: Step S4

The new accusation functions are obtained through an optimization under constraints, for a given estimated collusion $\hat{\boldsymbol{\theta}}$. We denote by μ_{Inn} and ν_{Inn} (resp. μ_{Coll} and ν_{Coll}) the expectation and variance of the distribution of the innocent users' scores (resp. colluders' scores), and $\kappa(S_j, S_k)$ the cross-correlation between scores of User j and User k. By construction of the code, the symbols are i.i.d from index to another. This and (1) implies that the statistics of the scores are linear with m:

$$\mu_{Inn} = m\tilde{\mu}_{Inn}, \qquad \nu_{Inn} = m\tilde{\nu}_{Inn}, \tag{2}$$

$$\mu_{Coll} = m\tilde{\mu}_{Coll}, \qquad \nu_{Coll} = m\tilde{\nu}_{Coll}. \tag{3}$$

The main constraints are now summed up:

- The scores of the innocents are centered: $\tilde{\mu}_{Inn} = 0$,
- The scores of the innocents are normalized so that $\tilde{\nu}_{Inn} = 1$,

- Two innocent users have independent scores, which under the Gaussian assumptions, amounts to set $\kappa(S_j, S_k) = 0$.

These are the same constraints as in Furon *et al*,[1] except that no consideration is paid to the variance of the colluders' scores.

The Kullback-Leibler distance measures the "distance" between the distributions of the colluders and innocents scores. Detection theory tells us that it should be as large as possible to make a clear cut between innocents and colluders, *i.e.* to ensure accurate verdicts. As we already considered that the scores of the colluders follow a Normal distribution $\mathcal{N}_{Coll}$, and the scores of innocent users a Normal distribution $\mathcal{N}_{Inn}$, the Kullback-Leibler distance between those two Normal distributions with $\tilde{\mu}_{Inn} = 0$ and $\tilde{\nu}_{Inn} = 1$ is as follows:

$$D_{KL}\left(\mathcal{N}_{Coll}, \mathcal{N}_{Inn}\right) = \frac{1}{2}\left(m\tilde{\mu}^2_{Coll} - \log(\tilde{\nu}_{Coll}) + \tilde{\nu}_{Coll} - 1\right). \tag{4}$$

As m is large, the prevailing term of the sum is $m\tilde{\mu}^2_{Coll}$. Our goal is thus to maximize $\tilde{\mu}_{Coll}$ under the constraints $\mu_{Inn} = 0$, $Cov(S_j, S_k) = 0$, and $\tilde{\nu}_{Inn} = 1$.

THEOREM 1. *Considering these conditions, the functions which maximize $\tilde{\mu}_{Coll}$ are*

$$g_{11}(p, \boldsymbol{\theta}) = \frac{1}{2\lambda}\frac{1-p}{q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}), \qquad g_{00}(p,\boldsymbol{\theta}) = \frac{1}{2\lambda}\frac{p}{1-q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}), \tag{5}$$

$$g_{10}(p,\boldsymbol{\theta}) = -\frac{p}{1-p}g_{11}(p,\boldsymbol{\theta}), \qquad g_{01}(p,\boldsymbol{\theta}) = -\frac{1-p}{p}g_{00}(p,\boldsymbol{\theta}), \tag{6}$$

with

$$\lambda = \frac{1}{2}\sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}, \tag{7}$$

$$q(p,\boldsymbol{\theta}) = \mathbb{P}(Y=1|P=p,\boldsymbol{\theta}), \tag{8}$$

$$A(p,\boldsymbol{\theta}) = \mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta}) - \mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta}). \tag{9}$$

These results allow us to compute the expression of the maximized μ_{Coll}:

$$\tilde{\mu}_{Coll} = \sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}. \tag{10}$$

Proof. We maximize this expression using the Lagrangian $J(g_{11}, g_{00}) = \tilde{\mu}_{Coll} - \lambda(\tilde{\nu}_{Inn} - 1)$. See details in appendix. □

2.3 Experimental results

For the experiments, we consider several strategies for the colluders to generate $\mathbf{Y}$:

- Uniform: the colluders randomly choose a symbol through their copies, therefore the probability they put a '1' is proportional to the number of '1' they got: $\mathbb{P}(Y=1|\Sigma=\sigma) = \sigma/c$;
- Majority: the colluders choose the most frequent symbol: $\mathbb{P}(Y=1|\Sigma=\sigma) = 1$ if $\sigma > c/2$, 0 else;
- Minority: the colluders choose the less frequent symbol: $\mathbb{P}(Y=1|\Sigma=\sigma) = 0$ if $\sigma > c/2$, 1 else;
- All1: If there is at least one '1', the colluders put a '1': $\mathbb{P}(Y=1|\Sigma=\sigma) = 1$ if $\sigma \neq 0$;
- All0: If there is at least one '0', the colluders put a '0': $\mathbb{P}(Y=1|\Sigma=\sigma) = 0$ if $\sigma \neq c$.

Moreover, the marking assumption always enforces that $\mathbb{P}(Y=1|\Sigma=0) = 0$ and $\mathbb{P}(Y=1|\Sigma=c) = 1$.

		Colluders' strategy				
	accusation strategy	Uniform	Majority	Minority	All1	All0
c=3	Uniform	**98 (71)**	106 (80)	100 (53)	97 (66)	97 (66)
	Majority	96 (67)	**110 (84)**	100 (34)	95 (59)	95 (59)
	Minority	81 (50)	59 (38)	**112 (75)**	89 (56)	89 (56)
	All1	83 (69)	88 (73)	88 (62)	**114 (68)**	84 (68)
	All0	83 (69)	88 (73)	88 (62)	84 (68)	**114 (68)**
c=4	Uniform	**98 (71)**	106 (80)	105 (44)	99 (62)	99 (62)
	Majority	96 (67)	**110 (84)**	105 (17)	97 (50)	97 (50)
	Minority	61 (34)	25 (15)	**128 (91)**	88 (53)	88 (53)
	All1	79 (65)	83 (63)	88 (72)	**121 (67)**	87 (67)
	All0	79 (65)	83 (63)	88 (72)	87 (67)	**121 (67)**
c=5	Uniform	**98 (71)**	110 (83)	110 (33)	100 (58)	100 (58)
	Majority	94 (63)	**120 (93)**	113 (-22)	98 (35)	98 (35)
	Minority	37 (19)	-20 (-17)	**155 (121)**	82 (52)	82 (52)
	All1	77 (59)	83 (47)	90 (90)	**128 (69)**	90 (69)
	All0	77 (59)	83 (47)	90 (90)	90 (69)	**128 (69)**

Table 1. Comparison of the values of $m\tilde{\mu}_{Coll}/\sqrt{\tilde{\nu}_{Inn}}$ obtained with our optimization of the accusation functions and, between brackets, those given by Furon *et al.*[1] Remind that Skoric's functions give a result of 64 for any colluders' strategy. The code length is $m = 100$ and the collusion sizes is $c = 3, 4, 5$. Expectations are in boldface font when the accusation function matches the collusion process.

2.3.1 Our optimization of the accusation functions is really efficient.

With these functions, we can compute the ratio $m\tilde{\mu}_{Coll}/\sqrt{\tilde{\nu}_{Inn}}$, the prevailing term in the Kullbach-Leibler distance for an arbitrary variance $\tilde{\nu}_{Inn}$. We compare it to the previous results of Furon *et al*,[1] in which $\tilde{\nu}_{Inn}$ was forced to be equal to 1 in any case, whereas here it is equal to 1 only when the accusation functions match the collusion strategy. Table 1 shows that the accusation fonctions are, as expected, more efficient for the colluders' strategy they were designed for. Moreover, when compared with the ones obtained in,[1] they are more efficient in any case, and behave better in mismatch cases.

2.3.2 A successful accusation, when usual Tardos code remains unsuccessful

First experiment: For the moment, the decoder knows c. Performances are uneven with respect to the collusion strategy as shown in Figure 2.3.2. It works great against 'All1', 'All0', 'Majority' and 'Minority'. However, the improvement when compared with Tardos is only mitigated against the 'Uniform' collusion strategy. This can be explained looking at Table 1. Our new functions have really greater expectations (increase of 100%) when dealing with 'All1', 'All0', 'Majority' and 'Minority', whereas the improvement is far less dramatic for the 'Uniform' strategy. This seems to prove that there are collusion process which are worse than others. This last comment is important as this fact is quite unobserved in the literature pertaining to Tardos codes. Indeed, Tardos original decoding performs equally whatever the decoding strategy, thus masking the fact that deterministic strategies are far less dangerous than probabilistic ones.

Second experiment: From now on, the decoder a priori ignores c. It is up to the EM algorithm to estimate the size of the collusion. The accuracy of this estimation appears to strongly depend on the ratio c/n. Figure 2.3.2 shows that if c is too small, the EM fails and the estimated strategy $\hat{\boldsymbol{\theta}}$ is not at all accurate. We can verify here that Tardos accusation process is independent of the colluders strategy whereas our algorithm strongly depends on it.

However, Figure 2.3.2 shows that when the collusion size is larger, the EM completes its role with precision and the performances of our decoding are much better than Tardos decoding, but they are very dependent on the collusion.

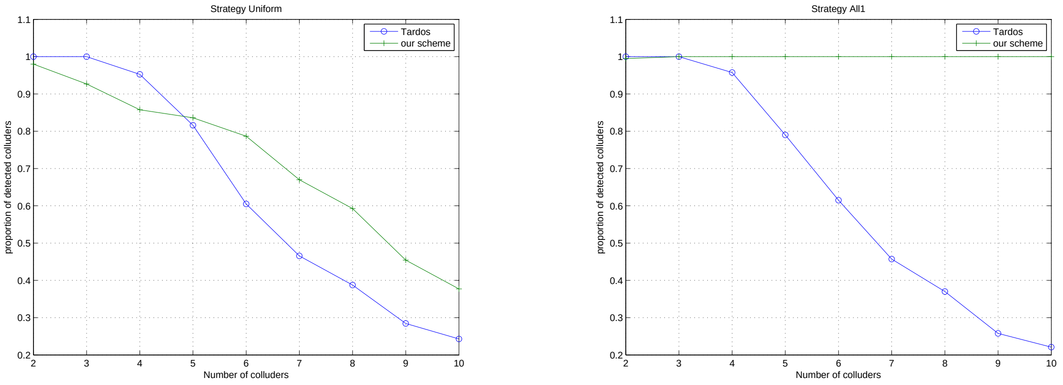


Figure 2. Comparison between Tardos decoding and our scheme for the 'Uniform' and 'All1' strategies. Proportion of caught colluders against the collusion size. $m = 1000$, $c \in \{2, \ldots, 10\}$, $n = 5000$.

Conclusion: We have found new accusation functions that are working much better than the original ones when matching the collusion strategy. The gap of performances is uneven. The remaining question is to find out why some collusion strategies are worse than others and, for a given size c, what is indeed the worst one. The iterative structure of our decoder allows to estimate the size and the strategy of the collusion. However, its efficiency is experimentally shown only when c is large, while small collusion sizes are still source of inaccuracy. Again, two fundamental questions raise: Are there collusion strategies harder than others to be estimated? What is the best estimator?

3. DESIGNING A SUITED WATERMARKING LAYER FOR MULTIMEDIA FINGERPRINTING

Any multimedia fingerprinting scheme must rely on a robust watermarking technique to embed users' codewords. We proposed a few months ago[3] a complete scheme based on Tardos codes for the fingerprinting layer, and the so-called "Broken Arrows" watermarking technique for the embedding layer. We experimented it on real images and showed this rationale leads to good robustness against collusion attacks: even in the case of averaging attacks, we are able to trace several colluders back, with a very small probability of error. "Broken Arrows"[6] is very robust: it has been intensively put to the test during BOWS-2 contest[7] and resisted well. Nevertheless, A. Westfeld designed during the contest a new and powerful attack[8] that may endanger any Wavelet-based watermarking technique. Our purpose in this section is to provide an improved version of "Broken Arrows" that is robust against this attack.

3.1 A rationale for a robust fingerprinting embedding scheme published at MM&SEC'08

We briefly recall in this section how we combined "Broken Arrows" with Tardos codes to obtain a complete fingerprinting scheme, so robust it can face the challenging fusion (or averaging) attacks.[3] We refer the reader to the original paper for more details and references.

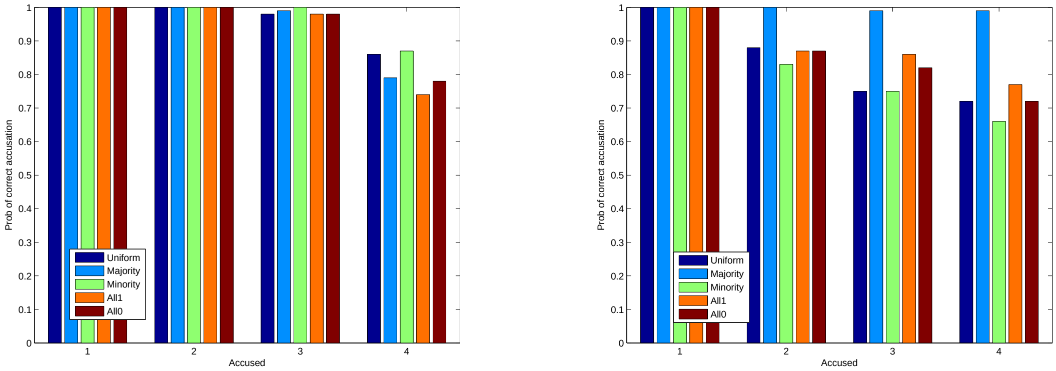


Figure 3. Comparison between Tardos decoding (on the left) and our scheme (on the right) for the 'Uniform', 'Majority', 'Minority', 'All1' and 'All0' strategies. Probability of correctly accusing the k-th highest score, for $k \in \{1, \ldots, 4\}$. $m = 1000$, $c = 4$, $n = 5000$, 100 experiments.

Among all the attacks the colluders may perform, we distinguish blocks exchange, fusion, and content processing (compression, filtering, de-noising, etc). The most challenging class in the design of a fingerprinting scheme is the fusion class, as general robustness is not *a priori* sufficient to resist such an attack. Our goal was to choose the right watermarking technique which, combined with Tardos codes, could face a fusion resulting of averaging copies of the multimedia content.

Our strategy was to enable the detection of several symbols simultaneously, each symbol corresponding to one of the averaged copies. Hence we needed a non-binary alphabet. We chose to use a very robust block based watermarking technique, and to embed one q-ary symbol in each block. As a *zero-bit* watermarking technique based on *on-off keying*, "Broken Arrows" can be adapted to permit such an embedding at low cost. By nature, a *zero-bit* technique does not carry any symbol, but just the presence of a watermark. To embed symbols of a q-ary alphabet, we defined q secret keys, and use key $K(X_{ji})$ to embed symbol X_{ji}. Since two different keys produce two almost independent watermark signals, the fusion is now deemed as a scaling and the addition of an independent noise. Note that adapting this way "Broken Arrows" to the q-ary case is not costly, algorithmically speaking: embedding and detection involve several nested spaces, and the secret key $K(X_{ji})$ is just related to the last one, so, the computational complexity overhead is only related to this last space.

We now give more details and introduce our notation, as we will need them to explain how we strengthen this scheme in the following sections. We first extract a long sequence $\mathbf{s}_o$ of L wavelet coefficients from the original content. This sequence is divided into blocks of l samples $\{\mathbf{s}_x^{(i)}\}_{i=1}^{m}$, such that $\mathbf{s}_x^{(i)} = (\mathbf{s}_o(il+1), \ldots, \mathbf{s}_o((i+1)l))$. We set l dividing L, and define $m = L/l$ as the length of the codewords to embed. The watermark embedding hides the symbol X_{ji} into the block $\mathbf{s}_x^{(i)}$, producing the i-th watermarked block:

$$\mathbf{s}_y^{(i)} = \mathbf{s}_x^{(i)} + \mathbf{w}(X_{ji}, \mathbf{s}_x^{(i)}),$$

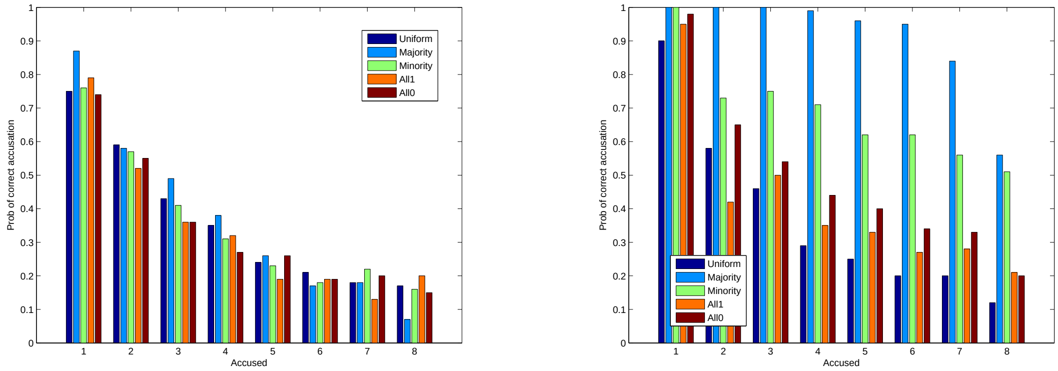


Figure 4. comparison between Tardos decoding (on the left) and our scheme (on the right) for the 'Uniform', 'Majority', 'Minority', 'All1' and 'All0' strategies. Probability of correctly accusing the k-th highest score, for $k \in \{1, \ldots, 8\}$. $m = 1000$, $c = 8$, $n = 5000$, 100 experiments.

where $\mathbf{w}(X_{ji}, \mathbf{s}_x^{(i)})$ denotes the embedded signal. Packing back all the watermarked blocks together, this yields the watermarked sequence $\mathbf{s}_{w,j}$, delivered to the j-th user.

3.2 Robustness enhancement of "Broken Arrows" to face Westfeld's attack

"Broken Arrows" (BA) has been designed to face a lot of usual attacks.[6] Nevertheless, it has been successfully attacked by A. Westfled during the first episode of BOWS-2 contest. Westfeld designed a specific attack for wavelet-based schemes, which can be regarded as a de-noising process.[8] It is mainly based on the estimation of the amplitude of any wavelet coefficient as a function of the coefficients in its neighbourhood *via* a regression. According to the author, this attack is successful because the watermark signal samples are independent whereas the wavelet coefficients of the host do not share this statistical property.

Hence, it seems that this attack could be prevented if the watermark signal has a similar dependency as the image itself. To strengthen BA's embedding, we propose two improvement directions: (i) Balancing the Wavelet Coefficients of three subbands in the same transformation level (BWC), and (ii) Averaging the Wavelet Coefficient with four neighbouring coefficients in the same subband (AWC). We will now briefly describe the original BA proportional embedding scheme before presenting our two improvements in the following sections.

3.2.1 The original BA proportional embedding

Why should we use a proportional embedding? We know that the traditional additive embedding has two major drawbacks. First, it does not comply with some psycho-visual basics, because its watermark signal power is constant over the entire image while the human eye is more sensitive on homogeneous regions than on textured regions and edges. The second drawback is that it does not respect the Power Spectrum Condition,[9] which states that the spectrum of the watermark has to be locally proportional to the spectrum of the host in order to be robust against de-noising attacks. The reason is that it is extremely hard or almost impossible to filter out the watermark signal if it shares the same statistical structure than the host.

A proportional embedding in the wavelet domain is proposed to solve this two issues. The watermarked signal can be represented as

$$\mathbf{s}_y^{(i)} = \mathbf{s}_x^{(i)} + \mathbf{mask}^{(i)}.\mathbf{w}^{(i)},$$

where $\mathbf{w}^{(i)}$ is the signal generated by the BA scheme in the wavelet domain according to the secret key $K(X_{ji})$, and **mask** denotes the perceptual mask that modulates it (*i.e.* **a**.**b** denotes the sample-wise multiplication of two vectors). In BA, the embedded signal is indeed proportional to the absolute value of the host wavelet coefficients:

$$\mathbf{mask_{BA}} = |\mathbf{s}_x^{(i)}|.$$

$|\mathbf{s}_x^{(i)}|$ denotes the absolute value of the wavelet coefficients of the host $\mathbf{s}_x^{(i)}$, which are the coefficients selected from all the bands except the low frequency LL band. Such an embedding in the wavelet domain provides a simple Human Visual System in the sense that it yields perceptually acceptable watermarked pictures for PSNR above 40 dB. In our experiments, presented in Section 3.3, we set the targeted PSNR = 43 dB as in the BOWS-2 contest.

With this PSNR, it appears that the amplitude of the samples of $\mathbf{w}^{(i)}$ are almost all lower than 1. Therefore, the BA technique conserves the sign of the wavelet coefficients. This, in our humble opinion, is the real security flaw of the technique: an attack not modifying the sign of the coefficients automatically preserves this important part of the original content. Therefore, if the amplitude of the attacked coefficients sufficiently different while preserving the quality of the content, the watermark can no longer be detected. This is indeed the case with the attack mounted by A. Westfeld.

3.2.2 BWC proportional embedding

Our first study consists in correlating coefficients of the three subbands in the same wavelet transformation level. In each level, we balance the wavelet coefficients of the three subbands. We denote by $\mathbf{mask}_{\mathbf{BWC}}^{(i)}(LH(k))$ (resp. $\mathbf{mask}_{\mathbf{BWC}}^{(i)}(HH(k))$ and $\mathbf{mask}_{\mathbf{BWC}}^{(i)}(HL(k))$) the sub-mask corresponding to subband $LH(k)$ (resp. $HH(k)$ and $HL(k)$). The Level 0 case is given as an example in the left hand side of Figure 5. These three sub-masks are all set to (we use $*$ to denote any subband LH, HH, HL):

$$\mathbf{mask}_{\mathbf{BWC}}^{(i)}(*(k)) = \frac{|\mathbf{s}_x^{(i)}(LH(k)) + \mathbf{s}_x^{(i)}(HH(k)) + \mathbf{s}_x^{(i)}(HL(k))|}{3},$$

where $\mathbf{s}_x^{(i)}(LH(k))$ (resp. $\mathbf{s}_x^{(i)}(HH(k))$ and $\mathbf{s}_x^{(i)}(HL(k))$) denotes the original wavelet coefficient of the host signal in subband $LH(k)$ (resp. $HH(k)$ and $HL(k)$); and $k \in \{0, 1, 2\}$ represents the wavelet decomposition scale. The mask $\mathbf{mask}_{\mathbf{BWC}}^{(i)}$ is the only difference between the BA and BWC proportional embedding methods.

Intuitively, this embedding enhances the dependency between the subbands of the wavelet coefficients of the watermark signal. For a given position, the mask has a value bigger than the amplitude of at least one of the three considered coefficients. Depending on the value of the watermark signal at this position, the embedding might consequently change the sign the wavelet coefficient. In other words, the presence of the watermark is not only hidden in the amplitudes of the coefficients but also in some of their signs. It is confirmed by our experimental results on Westfeld's de-noising attack.

3.2.3 AWC proportional embedding

Another avenue to improve the robustness of the embedding scheme is to take into account the dependency between the neighbouring coefficients. The main idea is inspired from Westfeld's attack. Our second study consists in replacing any coefficient of the wavelet transform by an average of five coefficients: itself and four neighbours. On the right hand side of Figure 5, we take the wavelet coefficients in the subband LH0 as an example. We obtain the mask

$$\mathbf{mask}_{\mathbf{AWC}}^{(i)}(m,n) = \frac{|\mathbf{s}_x^{(i)}(m-1,n) + \mathbf{s}_x^{(i)}(m,n-1) + \mathbf{s}_x^{(i)}(m,n) + \mathbf{s}_x^{(i)}(m+1,n) + \mathbf{s}_x^{(i)}(m,n+1)|}{5}.$$

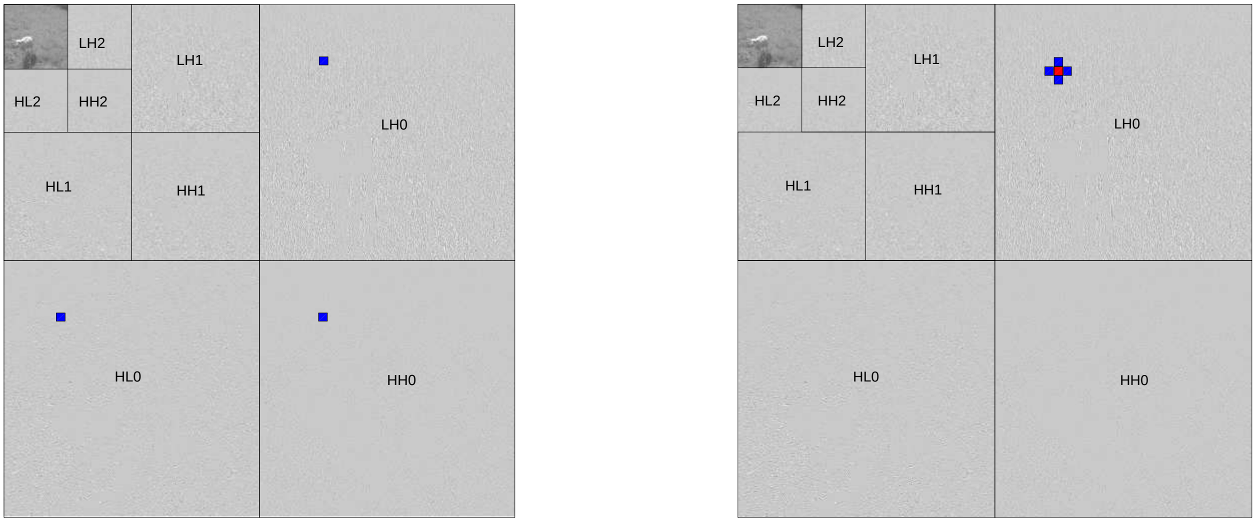


Figure 5. Balancing the Wavelet Coefficients of three subbands in each transformation level (BWC, left) and Averaging the Wavelet Coefficient with four neighbouring coefficients in the same subband (AWC, right)

$\mathbf{s}_x^{(i)}(m,n)$ denotes the wavelet coefficient in the position (m,n) for any band except the low frequency LL band. $\mathbf{s}_x^{(i)}(m-1,n)$, $\mathbf{s}_x^{(i)}(m,n-1)$, $\mathbf{s}_x^{(i)}(m+1,n)$, and $\mathbf{s}_x^{(i)}(m,n+1)$ are its four neighbours. Putting all the $\mathbf{mask}_{\mathbf{AWC}}^{(i)}(m,n)$ together, we get another mask $\mathbf{mask}_{\mathbf{AWC}}^{(i)}$, which will serve in the AWC proportional embedding. The masks $\mathbf{mask}_{\mathbf{AWC}}^{(i)}$, $\mathbf{mask}_{\mathbf{BWC}}^{(i)}$, $\mathbf{mask}^{(i)}$ are the only differences between the AWC, BWC and BA proportional embedding methods. In this way, the watermark signal modifies the signs of the host coefficients. As for BWC, this solution is sufficient to cope with Westfeld's attack. We will now detail our experimental results for both BWC and AWC.

3.3 Robustness evaluation

We first compare the robustness of BWC and AWC with the original BA, using the same benchmark as in the original paper.[6] Then we discuss the robustness according to Westfeld's attack.

3.3.1 Facing usual attacks: good results

We use the same $2,000$ luminance images of size $512\times\ 512$ as in.[6] These pictures represent natural and urban landscapes, people, or objects, taken with many different cameras from 2 to 5 millions of pixels. Three different watermark embedding strategies are compared: the original BA proportional embedding and the variants BWC and AWC. During embedding, the input PSNR is set to 43 dB; the real PSNR of the watermarked images is in between 42.5 dB and 43 dB. As for the original BA, the visual distortion are invisible for almost all images when using BWC or AWC. Indeed, these latter schemes yield slightly better quality (this is a subjective assessment).

We apply the same benchmark on watermarked images as in BA's original paper, that is, a number of attacks mainly composed of combinations of JPEG and JPEG 2000 compressions at different quality factors, low-pass filtering, wavelet subband erasure, and a simple de-noising algorithm. Figure 6 reveals the impact of 15 most significant attacks on the three embedding techniques. The probability of detecting the watermark (i.e. number of good detections divided by $2,000$) is plotted with respect to the average PSNR of the attacked images. Because these classical attacks produce almost the same average PSNR, the three points for a given attack are almost vertically aligned. The impact on the probability of detection is interesting: each watermark embedding technique has its advantage for resisting different attacks. AWC proportional embedding is more robust than others technique for resisting Attacks 9-14. BA proportional embedding is better for resisting Attacks 2, 5, and 6. For Attacks 1, 3, 4, 7, and 15, the three embedding technique have a comparable performance. Although the BWC proportional embedding has a tiny predominance for Attack 8, its overall performance is worse than other two techniques.

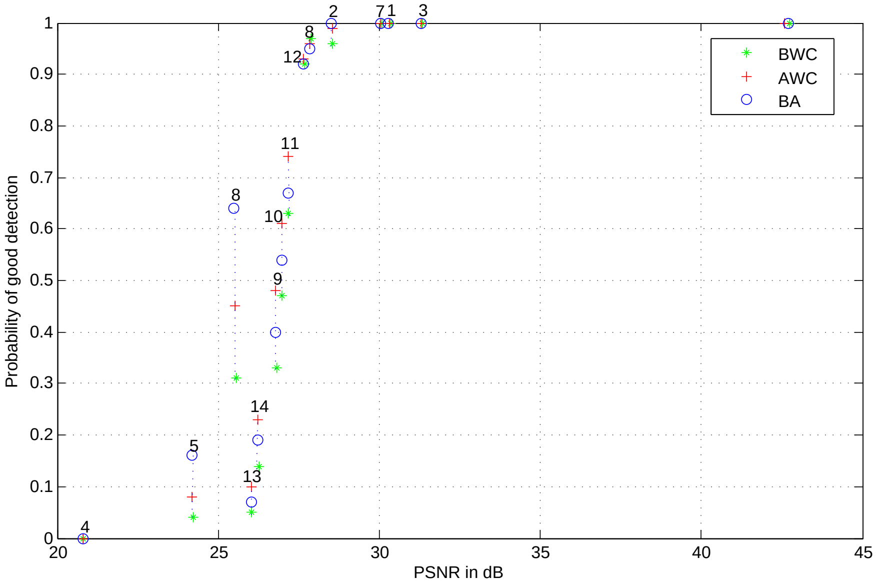


Figure 6. Probability of good detection versus average PSNR of the attacked images for the three watermark embedding techniques: 'BWC' proportional embedding '*', 'AWC' proportional embedding '+', 'BA' proportional embedding 'o'. Selection of attacks: 1) denoise threshold 20; 2) denoise threshold 30; 3) JPEG Q = 20; 4) JPEG2000 r = 0.001; 5) JPEG2000 r = 0.003; 6) JPEG2000 r = 0.005; 7) scale 1/2; 8) scale 1/3; 9) scale 1/3 + JPEG Q = 50; 10) scale 1/3 + JPEG Q = 60; 11) scale 1/3 + JPEG Q = 70; 12) scale 1/3 + JPEG Q = 90; 13) scale 1/4 + JPEG Q = 70; 14) scale 1/4 + JPEG Q = 80; 15) no attack.

3.3.2 Facing Westfeld's attack: a success

In this section, we evaluate the robustness of the three watermarking embedding techniques against Westfeld's attack. To get a result comparable with the above experiment, we keep the same testing conditions and use the same $2,000$ images. This is the main difference with Westfeld's test presented in,[8] as he used all the $10,000$ images available on BOWS-2's web site.

The PSNR of the attacked images ranges from 19.9 to 46.2 dB. This result is almost the same as Westfeld's (from 19.7 to 45.0 dB). Figure 7 shows the decreasing percentage of successfully broken images for increasing PSNR. For BA proportional embedding, Westfeld's attack is really powerful, successful for 100% of the images when the PSNR is less than 30 dB, and even if its efficiency decreases when the PSNR is growing, it is still successful for 40% of the images when the PSNR is around 35 dB. Note that this does not exactly fits Westfeld's experimental result, because he used a huger set of images. Nevertheless, our set is large enough to illustrate the power of its attack on the original BA.

Figure 7 also shows the results of both variants BWC and AWC. For BWC, Westfeld's attack does not work at all: the percentage of successfully broken images is 0 for any PSNR. For AWC, Westfeld's attack works for very few images for a PSNR ranging from 26 to 32 dB. Any of our two improved embedding techniques is sufficient to cope with Westfeld's attack. However, some robustness may be lost under some others common attacks, especially for the BWC proportional embedding. Therefore, in order to prevent Westfeld's attack as well as the others ones, we have to make a tradeoff, and the AWC proportional embedding seems to be the better choice. However, these enhancements might open other unforeseen security flaws. The cat and mouse game is not over...

Acknowledgement The authors would like to thank P. Bas for his constructive suggestions and inspiring discussions for the part of the watermark embedding.

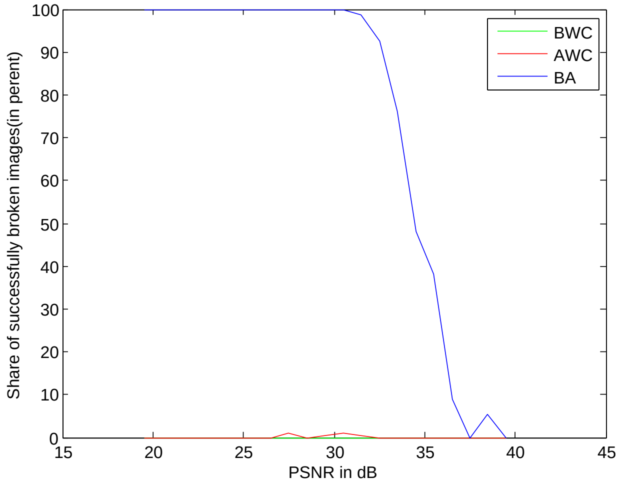


Figure 7. Operating curve of the estimated images from the BOWS2 database.

APPENDIX A. PROOF OF THEOREM 1

We want to find the functions $g_{11}(p,\boldsymbol{\theta})$, $g_{10}(p,\boldsymbol{\theta})$, $g_{01}(p,\boldsymbol{\theta})$ and $g_{00}(p,\boldsymbol{\theta})$ which maximize $\tilde{\mu}_{Coll}$ under the constraints $\tilde{\mu}_{Inn}=0$, $\kappa(S_j,S_k)=0$, and $\tilde{\nu}_{Inn}=1$.

First, let us compute all the data we need.

A.1 Statistics concerning innocent users

We set $q(p,\boldsymbol{\theta})=\mathbb{P}(Y=1|P=p,\boldsymbol{\theta})$, which can be expressed in terms of the collusion model:

$$q(p,\boldsymbol{\theta})=\sum_{\sigma=0}^{c}\mathbb{P}(Y=1|\Sigma=\sigma)\binom{c}{\sigma}p^{\sigma}(1-p)^{c-\sigma}.$$

Suppose first that the scores of the innocent users are centered. In this case, they are uncorrelated if

$$Cov(S_j,S_k)=\mathbb{E}_p\left[q(p,\boldsymbol{\theta})\left(pg_{11}(p,\boldsymbol{\theta})+(1-p)g_{10}(p,\boldsymbol{\theta})\right)^2+(1-q(p,\boldsymbol{\theta}))\left(pg_{01}(p,\boldsymbol{\theta})+(1-p)g_{00}(p,\boldsymbol{\theta})\right)^2\right]=0.$$

This leads to :

$$\begin{aligned} pg_{11}(p,\boldsymbol{\theta})+(1-p)g_{10}(p,\boldsymbol{\theta}) &= 0, && (11)\\ pg_{01}(p,\boldsymbol{\theta})+(1-p)g_{00}(p,\boldsymbol{\theta}) &= 0. && (12)\end{aligned}$$

Hence, functions g_{10} and g_{01} are determined by functions g_{11} and g_{00}. So, there are only two functions left to be optimized.

Can we get more constraints on g_{00} and g_{11} from the relation $\tilde{\mu}_{Inn}=0$? By definition, we have

$$\tilde{\mu}_{Inn}=\mathbb{E}_p\left[q(p,\boldsymbol{\theta})\left(pg_{11}(p,\boldsymbol{\theta})+(1-p)g_{10}(p,\boldsymbol{\theta})\right)+(1-q(p,\boldsymbol{\theta}))\left(pg_{01}(p,\boldsymbol{\theta})+(1-p)g_{00}(p,\boldsymbol{\theta})\right)\right].$$

But, due to Relations (11) and (12), the right handside term is equal to zero. Hence, we do not get any new relation between g_{00} and g_{11}. The variance has a similar expression but with square of functions, and get simplified using (11) and (12):

$$\begin{aligned}\tilde{\nu}_{Inn} &= \mathbb{E}_p\left[q(p,\boldsymbol{\theta})\left(pg_{11}^2(p,\boldsymbol{\theta})+(1-p)g_{10}^2(p,\boldsymbol{\theta})\right)+(1-q(p,\boldsymbol{\theta}))\left(pg_{01}^2(p,\boldsymbol{\theta})+(1-p)g_{00}^2(p,\boldsymbol{\theta})\right)\right], && (13)\\ &= \mathbb{E}_p\left[q(p,\boldsymbol{\theta})\frac{p}{1-p}g_{11}^2(p,\boldsymbol{\theta})+(1-q(p,\boldsymbol{\theta}))\frac{1-p}{p}g_{00}^2(p,\boldsymbol{\theta})\right]. && (14)\end{aligned}$$

A.2 Statistics concerning colluders

Let us simplify the expression of the mean of the colluders scores:

$$\begin{aligned}\tilde{\mu}_{Coll} &= \mathbb{E}_p[p(\mathbb{P}(Y=0|X=1,P=p,\boldsymbol{\theta})g_{01}(p,\boldsymbol{\theta})+\mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta})g_{11}(p,\boldsymbol{\theta}))\\ &+ (1-p)(\mathbb{P}(Y=0|X=0,P=p,\boldsymbol{\theta})g_{00}(p,\boldsymbol{\theta})+\mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta})g_{10}(p,\boldsymbol{\theta})]. \end{aligned} \quad (15)$$

Using Relations (11) and (12), and the properties

$$\begin{aligned}\mathbb{P}(Y=0|X=0,P=p,\boldsymbol{\theta})+\mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta}) &= 1\\ \mathbb{P}(Y=0|X=1,P=p,\boldsymbol{\theta})+\mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta}) &= 1 \end{aligned} \quad (16)$$

we get

$$\begin{aligned}\tilde{\mu}_{Coll} &= \mathbb{E}_p\left[(\mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta})-\mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta}))(pg_{11}(p,\boldsymbol{\theta})+(1-p)g_{00}(p,\boldsymbol{\theta}))\right],\\ &= \mathbb{E}_p\left[A(p,\boldsymbol{\theta})\left(pg_{11}(p,\boldsymbol{\theta})+(1-p)g_{00}(p,\boldsymbol{\theta})\right)\right], \end{aligned} \quad (17)$$

with $A(p,\boldsymbol{\theta})=\mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta})-\mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta})$. These terms are calculated as follows:

$$\mathbb{P}(Y=1|X=1,p,\boldsymbol{\theta}) = \sum_{\sigma=1}^{c}\mathbb{P}(Y=1|\Sigma=\sigma)\binom{c-1}{\sigma-1}p^{\sigma-1}(1-p)^{c-\sigma}, \quad (18)$$

$$\mathbb{P}(Y=1|X=0,p,\boldsymbol{\theta}) = \sum_{\sigma=0}^{c-1}\mathbb{P}(Y=1|\Sigma=\sigma)\binom{c-1}{\sigma}p^{\sigma}(1-p)^{c-\sigma-1}. \quad (19)$$

A.3 Lagrangian

We want to maximize $\tilde{\mu}_{Coll}$ under the constraint that $\tilde{\nu}_{Inn}=1$, using a Lagrangian:

$$J(g_{11}(p,\boldsymbol{\theta}),g_{00}(p,\boldsymbol{\theta})=\tilde{\mu}_{Coll}-\lambda(\tilde{\nu}_{Inn}-1).$$

This functional depends on functions and reaches an extrenum when a small perturbation of the inputs functions does not change its value. We define a derivative with respect to a function as the linear term in the Taylor expansion: $J(g_{11}(p,\boldsymbol{\theta})+\epsilon(p),g_{00}(p,\boldsymbol{\theta}))=J(g_{11}(p,\boldsymbol{\theta}),g_{00}(p,\boldsymbol{\theta}))+\frac{\partial J(g_{11}(p,\boldsymbol{\theta}),g_{00}(p,\boldsymbol{\theta}))}{\partial g_{11}(p,\boldsymbol{\theta})}+\mathbb{E}_p[o(\epsilon(p))]$.

$$\begin{aligned}\frac{\partial J(g_{11}(p,\boldsymbol{\theta}),g_{00}(p,\boldsymbol{\theta}))}{\partial g_{11}(p,\boldsymbol{\theta})} &= \mathbb{E}_p\left[pA(p,\boldsymbol{\theta})\epsilon(p)\right]-\lambda\mathbb{E}_p\left[2\frac{p}{1-p}q(p,\boldsymbol{\theta})g_{11}(p,\boldsymbol{\theta})\epsilon(p)\right]\\ &= \mathbb{E}_p\left[p\epsilon(p)\left(A(p,\boldsymbol{\theta})-2\lambda\frac{q(p,\boldsymbol{\theta})}{1-p}g_{11}(p,\boldsymbol{\theta})\right)\right].\end{aligned}$$

This equals zero for any $\epsilon(p)$ if

$$g_{11}(p,\boldsymbol{\theta})=\frac{1}{2\lambda}\frac{1-p}{q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}). \quad (20)$$

Cancel the derivate with respect to the second function, we obtain

$$g_{00}(p,\boldsymbol{\theta})=\frac{1}{2\lambda}\frac{p}{1-q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}), \quad (21)$$

with λ given by the constraint:

$$\lambda=\frac{1}{2}\sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}. \quad (22)$$

Plugging (21) and (20) in (17), we get:

$$\tilde{\mu}_{Coll}=\sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}. \quad (23)$$

REFERENCES

[1] T.Furon, A.Guyader, and F.Cerou, "On the design and optimization of tardos probabilistic fingerprinting codes," *Information Hiding 2008, Santa Barbara, California, USA* (May 2008).

[2] G.Tardos, "Optimal probabilistic fingerprint codes," *Proc. of the 35th annual ACM symposium on theory of computing* , 116–125 (2003).

[3] F.Xie, T.Furon, and C.Fontaine, "On-off keying modulation and tardos fingerprinting," *Proc. of 10th ACM Multimedia and Security Workshop, Oxford, UK* (September 2008).

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

[5] T.Furon, A.Guyader, and F.Cerou, "Experimental assessment of the reliability for watermarking and fingerprinting schemes," *accepted to EURASIP Journal on Information Security* (2008).

[6] T.Furon and P.Bas, "Broken arrows," *accepted to EURASIP Journal on Information Security* (2008).

[7] BOWS-2, "http://bows2.gipsa-lab.inpg.fr/index.php?mode=view&tmpl=index1," (2007).

[8] A.Westfeld, "A regression-based restoration technique for automated watermark removal," *Proc. of 10th ACM Multimedia and Security Workshop, Oxford, UK* (September 2008).

[9] Su, J., J. Eggers, and B. Girod, "Analysis of digital watermarks subjected to optimum linear filtering and additive noise," *Signal processing* **81**, 1141–1175 (2001).