Enhanced Blind Decoding of Tardos Codes with New Map-Based Functions

Mathieu Desoubeaux, Cédric Herzet, William Puech, Gaetan Le Guelvouit

ENHANCED BLIND DECODING of TARDOS CODES with NEW MAP-BASED FUNCTIONS

Mathieu Desoubeaux [#1], Cédric Herzet [*2], William Puech [#], Gaëtan Le Guelvouit [o]

[#] *University of Montpellier 2 - LIRMM - 34095 - Montpellier Cedex 5 - France*
[*] *INRIA - Centre Rennes - Bretagne Atlantique - Equipe Fluminance - 35000 - Rennes - France*
[o] *Orange Labs - 35512 - Cesson-Sévigné - France*
[1] mathieu.desoubeaux@lirmm.fr
[2] cedric.herzet@inria.fr

***Abstract*—This paper presents a new decoder for probabilistic binary traitor tracing codes under the marking assumption. It is based on a binary hypothesis testing rule which integrates a collusion channel relaxation so as to obtain numerical and simple accusation functions. This decoder is blind as no estimation of the collusion channel prior to the accusation is required. Experimentations show that using the proposed decoder gives better performance than the well-known symmetric version of the Tardos decoder for common attack channels.**

I. Introduction

Active fingerprinting, also known as traitor tracing, first introduced in [1], aims at finding the leak of an illegal redistribution of copyrighted digital contents. This goal requires to personalize each delivered content by embedding a sequence of symbols associated to each user.

Recent trends to generate such sequences focus on probabilistic codes since they allow for low error probabilities (namely the sum of the false alarm probability and the false negative probability) with affordable code lengths and small alphabet's sizes. The performance of the probabilistic codes is usually measured in terms of the minimum code length required to achieve a given error probability.

One of the most efficient probabilistic codes have been proposed by Tardos in [3]. These codes rely on the so-called "marking assumption", first introduced by Boneh and Shaw in [4]. In particular, the Tardos code was the first one proposed in the literature whose length, say m, scales as $\mathcal{O}(c^2 \ln(\epsilon_1^{-1}))$, where ϵ_1 represents the false alarm probability and c is the maximum number of colluders. Tardos code gets therefore very close to the lower bound on the code length proved in [5] and [3], which states that $m = \Omega(c^2 \ln(\epsilon_1^{-1}))$ for random codes and any number of users n with $n \geq c + 1$ users.

Since Tardos seminal work, many efforts have been devoted to further improve the efficiency and the effectiveness of his code. First, the authors of [6], [7] aimed at reducing the constant appearing in the code-length bound. In [8], [9], the authors respectively focussed on the improvement of the memory consumption and the decoding complexity. Finally, other contributions [10], [11] addressed the problem of characterizing probabilistic codes in terms of achievable capacity from an information-theoretical point of view. The latter results have led to several improvements of the decoding functions, see [12], [13].

In this paper, we are concerned with "simple" decoders such as the original one proposed by Gabor Tardos which provides a theoretical proof of performance under threshold-based decisions. Such decoders have a decoding complexity scaling as $O(n)$. Joint decoders, requiring to analyze subsets of (up to) c possible traitors among n users lead to better performance [11] but for higher complexity. This type of decoder will therefore be out of the scope of this paper.

The original "simple" Tardos decoder is known to be suboptimal if the collusion channel, *i.e.,* the coalition strategy and the coalition size, is known at the decoding side. However as it is unknown in practice, an approach given in [13] solved this problem with an estimated collusion channel. Even though better effectiveness is achieved for small coalition, this approach remains complex for large coalitions. Additionnaly the bound on the false alarm probability is not ensured as the authors of [13] have not solved the threshold based decision issue.

As the original Tardos decoder, our decoder is "agnostic" because it does not need to estimate the collusion channel. Our decoder addresses the traitor tracing problem under the marking assumption and solves a test under a specific Maximum a Posteriori (MAP) decision rule. In particular, the decision rule is devised under the assumption that the densities of probability of both the traitors strategy and the coalition size follow a *non-informative* law.

We compare our decoder with the symmetric version of the Tardos decoder given in [7]. Generating the receiver operating characteristic (ROC) by Monte Carlo simulation, we show that better result can be obtained with the proposed methodology in all the considered settings. In particular, the efficiency of our proposal is presented for common collusion

MMSP'13, Sept. 30 - Oct. 2, 2013, Pula (Sardinia), Italy.
???-?-????-????-?/10/$??.??

channels presented in the literature and different code lengths. The improvement of the performance is however at the cost of a small increase of the decoding complexity. Indeed the complexity of our decoder scales as $\mathcal{O}(nc)$ and, unlike Tardos decoding, varies therefore linearly with the maximum number of colluders.

The rest of the paper is organized as follows: Section 2 provides the probabilistic model of the problem. Section 3 presents the rationale of our decoding approach and the details of the proposed decoder. Section 4 is concerned with the experimental evaluation. Finally Section 5 gives some concluding remarks.

II. Notations

Throughout the paper, we will use the following notations. We use uppercase letters for random variables, lowercase letters for their individual values, and boldface fonts for sequences (or vectors). $\mathbb{P}_X(x)$ will denote the probability of random variable X evaluated at x. However, when there is no possible ambiguity, we often use the shorthand notation: $\mathbb{P}(x) \triangleq \mathbb{P}_X(x)$. The binomial coefficient indexed by n and k is denoted $\binom{n}{k}$.

III. Probabilistic Model

Let $\mathbf{X} \in \{0,1\}^{m\times n}$ define a length-m binary code for n users. In practical systems, a different column of the code $\mathbf{X}$ denoted $\mathbf{x_j}$ is hidden in the multimedia content delivered to each user j. We assume that c users (referred to as the colluders) combine their contents to form a new sequence $\mathbf{y} \in \{0,1\}^m$. In the sequel, we will identify the users participating to the collusion by a vector $\mathbf{s} \in \{0,1\}^n$ defined as follows: $s_j = 1$ if the jth user is a colluder and $s_j = 0$ otherwise. Clearly, we have the following relation between c and $\mathbf{s}$:

$$c = \sum_i s_i.$$

For a given size of collusion c, we assume that all the repartition of the colluders within the users are equally likely, that is

$$\mathbb{P}(\mathbf{s}|c) = 1/\binom{n}{c}. \tag{1}$$

Moreover, it is commonly assumed that the ith element of $\mathbf{y}$ only depends on the number of 1's appearing in the colluder's codewords at position i. More formally, let $\mathbf{t} \in \{0,\dots,c\}^m$ be a vector whose ith element is the number of symbols "1" in the colluder sequences at position i, $1 \le i \le m$. We have therefore

$$\mathbf{t} = \mathbf{X}\mathbf{s}. \tag{2}$$

Given $\mathbf{t}$, the probability of the sequence $\mathbf{y}$ generated by the colluders is totally characterized by the following conditional probability

$$\mathbb{P}(\mathbf{y}|\mathbf{t},\mathbf{G}) = \prod_i \mathbb{P}(y_i|t_i,\mathbf{G}), \tag{3}$$

where

$$\mathbb{P}_{Y_i|T_i,G}(y_i|t_i = k,\mathbf{G}) \sim Ber(g_{ki}).$$

Hence, the choice of the matrix $\mathbf{G}$ of Bernoulli parameters fully characterizes the collusion strategy of a coalition of size c. For clarity, we do not specify the parameter c in the notation of the matrix $\mathbf{G}$ of size $m \times c$. The elements of $\mathbf{G}$ can be arbitrary except for the elements g_{ik} with $k \in \{0, c\}$ which should obey the so-called "marking assumption" [4], that is $g_{i0} = 0$ and $g_{ic} = 1\ \forall i$.

In practice, the ability of any system to identify the colluders (*i.e.,* the vector $\mathbf{s}$) from the observed sequence $\mathbf{y}$ strongly depends on the code $\mathbf{X}$ used to protect the content. In his seminal paper [3], Tardos proposed to construct the code in a probabilistic manner as follows:

$$\mathbb{P}(\mathbf{X}|\mathbf{p}) = \prod_{i=1}^{m}\prod_{j=1}^{n} \mathbb{P}(x_{ij}|p_i), \tag{4}$$

where

$$\mathbb{P}_{X_{ij}|P_i}(x_{ij}|p_i) \sim Ber(p_i),$$

and $\mathbf{p}$ denotes the secret vector collecting the Bernoulli parameters p_i. Moreover, Tardos proposed a specific distribution to generate the latter parameters:

$$\mathbb{P}(\mathbf{p}) = \prod_{i=1}^{m} \mathbb{P}(p_i), \tag{5}$$

with[1]

$$\mathbb{P}_{P_i}(p_i) \sim (1/(\pi\sqrt{p_i(1-p_i)})), \text{ with } p_i \in]0,1[.$$

In conclusion, we have that the joint probability of the different quantities entering into play in the conception of the observed sequence $\mathbf{y}$ by the colluders defined in $\mathbf{s}$ can be expressed as follows:

$$\mathbb{P}(\mathbf{y},\mathbf{t},\mathbf{X},\mathbf{p},\mathbf{s}|c,\mathbf{G}) = \mathbb{P}(\mathbf{y}|\mathbf{t},\mathbf{G})\mathbb{P}(\mathbf{t}|\mathbf{X},\mathbf{s})\mathbb{P}(\mathbf{s}|c)\mathbb{P}(\mathbf{X}|\mathbf{p})\mathbb{P}(\mathbf{p}), \tag{6}$$

where the different conditional probabilities appearing in the right-hand side of (6) have been defined in (1), (2), (3), (4) and (5). In the next section, we will exploit this sound probabilistic characterization of the system to derive a new colluder detector.

IV. Decoding Description

The ultimate goal of any fingerprinting system is to accurately identify the users responsible of the release of the pirated content. More formally, this requires to properly estimate the "accusation" vector $\mathbf{s}$ from the observed sequence $\mathbf{y}$. In practice, it is sufficient to accuse at least one guilty user while innocent users are deemed guilty with sufficiently low probability. This task often results in a compromise between accuracy and computational complexity. This section is dedicated to the derivation of a novel decoder offering a good

[1]We omit here the cutoff parameter for the sake of simplicity.

trade-off between these two contradicting goals. In section IV-A, we first replace our contribution in the existing literature. Then, in section IV-B, we derive the accusation functions defining our decoder.

A. Connections with previous contributions

This section is dedicated to linking our approach with existing "simple" decoders. The identification of such decoders is related to a score σ_j which gives sufficient information about the involvement of a user j in the forgery of $\mathbf{y}$. The sources of information available at the decoder for the evaluation of one user's score are the forgery $\mathbf{y}$, the sequence of the user $\mathbf{x_j}$ and the secret vector $\mathbf{p}$. In such a context, in order to prevent accusation of innocent users, two scenarios are possible: either all users with a score above a threshold are accused or only the user with the biggest score above the threshold is accused. The decoder is evaluated in terms of soundness and completeness. The decoder is said to be ϵ_1-sound if the false alarm probability is bounded by ϵ_1, and said to be ϵ_2-complete if the false negative probability is bounded by ϵ_2 for a maximum coalition size.

Two kinds of simple decoders exist. The first ones adapt their scores computation to the collusion channel as in [13]. The worst case attacks are still unknown for such decoders and the false alarm probability is not bounded for any coalition sizes. Their effectiveness is experimentally assessed. On the contrary, the second class of decoders is independent of the collusion channel as in [3]. Theoretical proofs of soundness and completeness are given in [3] by using Chernoff bounds.

In an informed setup, where the decoder knows the collusion channel and the size of the collusion, the Neyman-Pearson theorem tells us that the optimal discriminative score, say σ_j^{NP}, to test whether user j pertains to the collusion or not is as follows:

$$\sigma_j^{\mathrm{NP}} = \frac{\mathbb{P}(\mathbf{y}|\mathbf{x}_j, s_j = 1, \mathbf{G}, \mathbf{p}, c)}{\mathbb{P}(\mathbf{y}|\mathbf{x}_j, s_j = 0, \mathbf{G}, \mathbf{p}, c)}. \tag{7}$$

However this scoring is out of reach since the collusion channel is unknown to the decoder. Some class of agnostic decoders exist where scoring functions are independent of the collusion size c and the collusion strategy $\mathbf{G}$. In [14], the authors proposed a symmetric version of the original Tardos approach. This decoder computes a score for each user as

$$\sigma_j^t = \sum_{i=1}^{m} U(y_i, x_{ij}, p_i), \tag{8}$$

with

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

and

$$U(1,0,p_i) = -U(1,1,p_i), \quad U(0,1,p_i) = -U(0,0,p_i).$$

This scoring function ensures some kind of separation between the distributions of scores of the innocent and traitor users for any collusion channel compliant with the marking assumption. It then permits to derive an appropriate threshold which guarantees to bound the false alarm probability for a given code length m.

However the authors of [13] have shown the huge gap between the symmetric Tardos decoder and the informed decoder of equation (7). Hence, as the collusion channel is unknown in practice, they have proposed to estimate it. Their approach is based on the so-called Expectation-Maximization algorithm. The authors assume that the collusion strategy is constant for all positions i for the sake of simplifying the mathematical model. Our assumption on the collusion channel is more general in the sense that our decoder considers all possible strategies at each ith position.

B. MAP Decoding with Non-informative Priors

The challenge of robust and effective detection procedures stands in the fact that some parameters (namely $\mathbf{G}$ and c) of the model are actually unknown to the decoder. Now, a very common approach in Bayesian statistics consists in defining non-informative priors on the unknown quantities and marginalize them out from the joint probability characterizing the system. By "non-informative" prior, it is usually understood a probability distribution not favoring any of the possible realizations of the considered random variable.

More specifically, our approach consists in exploiting the following joint probability to derive our decoder:

$$\mathbb{P}(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{p}, \mathbf{s}) = \sum_c \left(\int \mathbb{P}(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{p}, \mathbf{s}|c, \mathbf{G}) \mathbb{P}(\mathbf{G}) d\mathbf{G} \right) \mathbb{P}(c),$$

where $\mathbb{P}(\mathbf{y}, \mathbf{t}, \mathbf{X}, \mathbf{p}, \mathbf{s}|c, \mathbf{G})$ has been specified in (6) and $\mathbb{P}(c)$, $\mathbb{P}(\mathbf{G})$ are non-informative priors which will be defined hereafter. In turn, these joint probabilities can be marginalized to compute the following likelihood ratio

$$\sigma_j^{\mathrm{MAP}} \triangleq \frac{\mathbb{P}(s_j = 1|\mathbf{y}, \mathbf{x}_j, \mathbf{p})}{\mathbb{P}(s_j = 0|\mathbf{y}, \mathbf{x}_j, \mathbf{p})} = \frac{\mathbb{P}(\mathbf{y}, \mathbf{x}_j, \mathbf{p}, s_j = 1)}{\mathbb{P}(\mathbf{y}, \mathbf{x}_j, \mathbf{p}, s_j = 0)}.$$

At the decoding side, for a given code length, it is illusive to chase more than say $c_{\max}$ colluders. The size of the collusion is seen as a discrete random variable ranging from 1 to $c_{\max}$. Therefore, its non-informative prior distribution is the uniform law, that is

$$\mathbb{P}(c) = \frac{1}{c_{\max}}. \tag{9}$$

As for the collusion channels, we first assume the statistical independence of the parameters:

$$\mathbb{P}(\mathbf{G}) = \prod_{i,k} \mathbb{P}(g_{ik}).$$

We then enforce the marking assumption: $g_{i0} = 1 - g_{ic} = 0$, $\forall i$. The other parameters G_{ik} are seen as continuous random variables ranging in $[0, 1]$. Not favoring any of the possible realizations of G_{ik}, we set the intuitive uniform law as a non-informative prior distribution. Then marginalizing over g_{ik} for

$k \in \{1, ..., c-1\}$ leads to:

$$\mathbb{P}(y_i|t_i = k) = \int_0^1 \mathbb{P}(g_{ik}) g_{ik}^y (1-g_{ik})^{1-y} d_{g_{ik}} \quad (10)$$

$$= \int_0^1 g_{ik}^y (1-g_{ik})^{1-y} d_{g_{ik}} = 1/2. \quad (11)$$

Notice that all Beta law Beta(α,β) of equal parameters, such as the well known Jeffreys prior of parameters $\alpha = \beta = 1/2$, give an equivalent result as in (11). Indeed if

$$P(g_{ik}) = \frac{g_{ik}^{\alpha-1}(1-g_{ik})^{\beta-1}}{B(\alpha,\beta)}, \quad (12)$$

with

$$B(\alpha,\beta) = \int_0^1 v^{\alpha-1}(1-v)^{\beta-1} dv$$

and if $\alpha = \beta$,

$$\mathbb{P}(y_i|t_i = k) = 1/2. \quad (13)$$

The uniform law is then just a Beta law of parameters $\alpha = \beta = 1$ in (12).

Let us note that if the realizations of c and $\mathbf{G}$ obeyed probabilities $\mathbb{P}(c)$ and $\mathbb{P}(\mathbf{G})$, c- and $\mathbf{G}$-blind optimal Neyman-Pearson test would result in a simple thresholding of σ_j^{MAP}.

Let us then particularize the expression of σ_j^{MAP} to the particular hypotheses introduced in (9) and (13). We have

$$\sigma_j^{\text{MAP}} = \frac{\sum_{c=1}^{c_{max}} \mathbb{P}(\mathbf{y}|s_j = 1, \mathbf{x_j}, c)\mathbb{P}(s_j = 1|c)}{\sum_{c=1}^{c_{max}} \mathbb{P}(\mathbf{y}|s_j = 0, \mathbf{x_j}, c)\mathbb{P}(s_j = 0|c)}, \quad (14)$$

where

$$\begin{aligned} \mathbb{P}(s_j = 1|c) &= c/n, \\ \mathbb{P}(s_j = 0|c) &= (n-c)/n, \end{aligned}$$

$$\begin{aligned} \mathbb{P}(\mathbf{y}|s_j, \mathbf{x}_j, c) &= \prod_i \mathbb{P}(y_i|s_j, x_{ij}, c) \\ &= \prod_i \sum_{t_i=0}^{c} \mathbb{P}(y_i, t_i|s_j, x_{ij}) \\ &= \prod_i \sum_{t_i=0}^{c} \mathbb{P}(y_i|t_i)\mathbb{P}(t_i|s_j, x_{ij}, c) \end{aligned}$$

and

$$\mathbb{P}(t_i|s_j = 1, x_{ij}, c) = \binom{c-1}{t_i - x_{ij}} p_i^{t_i - x_{ij}} (1-p_i)^{c-1-t_i+x_{ij}},$$

$$\mathbb{P}(t_i|s_j = 0, x_{ij}, c) = \binom{c}{t_i} p_i^{t_i} (1-p_i)^{c-t_i}.$$

Particularizing these expressions to (13), we obtain after some algebraic manipulations:

$$\begin{aligned} \mathbb{P}(y_i|s_j = 1, x_{ij}) &= 1/2 \times (1 + (-1)^{y_i}((1-x_{ij}) \\ &\times (1-p_i)^{c-1} - x_{ij} p_i^{c-1})), \\ \mathbb{P}(y_i|s_j = 0, x_{ij}) &= 1/2 \times (1 + (-1)^{y_i}((1-p_i)^c - p_i^c)). \end{aligned}$$

C. Numerical solution

The evaluation of the large products appearing in equation (14) suffers from numerical problems w.r.t. machine finite precision. The logarithm translates products into sums. However taking the logarithm of our test does not give a simple formulation due to the sum over the possible coalition sizes. We resort to generalized maximum function M_g as shown in [15]:

$$\begin{aligned} M_g(a,b) &\triangleq \log(\exp(a) + \exp(b)) \\ &= \max(a,b) + \log(1 + e^{-|a-b|}), \\ M_g(a,b,c) &\triangleq \log(\exp(a) + \exp(b) + \exp(c)) \\ &= M_g\left(M_g(a,b), c\right). \end{aligned}$$

The test (14) can be formulated as follows in the logarithmic domain:

$$\sigma_j^{\text{MAP}} = \log\left(\sum_{c=2}^{c_{max}} e^{A1_c}\right) - \log\left(\sum_{c=2}^{c_{max}} e^{A2_c}\right),$$

with

$$\begin{aligned} A1_c &= \log\frac{c}{n} + \sum_{i=1}^{m} \log \mathbb{P}(y_i|s_j = 1, x_{ij}, c), \\ A2_c &= \log\frac{n-c}{n} + \sum_{i=1}^{m} \log \mathbb{P}(y_i|s_j = 0, x_{ij}, c). \end{aligned}$$

The generalized maximum function gives the following recursive expression of the test:

$$\sigma_j^{\text{MAP}} = M_g(M_g(...), A1_{c_{max}}) - M_g(M_g(...), A2_{c_{max}}).$$

V. Simulation Results

The experimental investigation is composed of two parts. The first one presents the effectiveness of the method for different collusion channels. The second part presents the effectiveness of the method for different code lengths. We used the classical Monte Carlo estimator to estimate the performance.

We compared our approach with two decoders. The first decoder is the symmetric version of the Tardos decoder given in (8). The second decoder is the informed decoder given in (7).

Receiver Operating Characteristics (ROC) curves plots are used to compare the three decoders. Deterministic or *random* strategies are considered as follows. "Minority" and "Majority" are deterministic strategies where the less or the most frequent symbol is put in the pirated sequence. "Uniform", "Coin flip" and 'Worst case attack" are random strategies. In the "Uniform" strategy, the colluders uniformly-randomly choose one of their symbols. In the "Coin flip" strategy, the colluders flip a fair coin to choose a symbol. Finally in the "Worst case attack" (wca) strategy, the colluders minimize the mutual information between the symbols of the pirated copy and each of their codewords. It is considered to be the worst attack against the best achievable simple decoder from an information-theoretical viewpoint. This strategy is obtained by

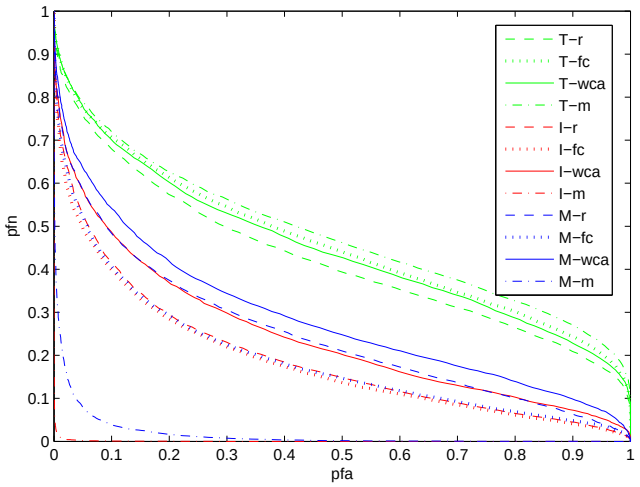

Fig. 1. ROC curves of the informed decoder, the MAP blind decoder and the symmetric Tardos decoder for 4 different collusion channels with $m = 300$, $c = 6$ and $n = 1000$ users.

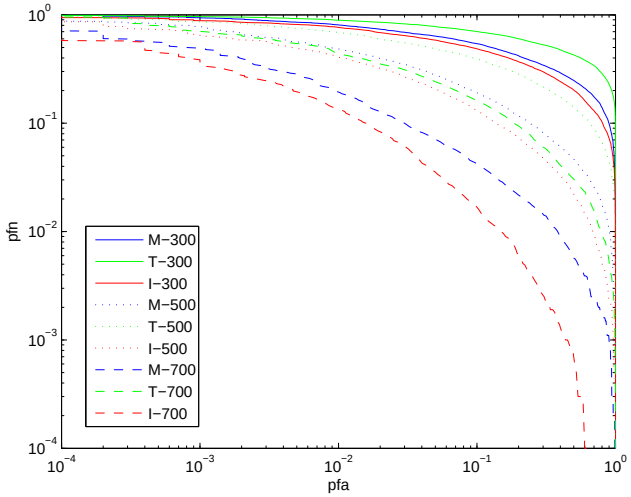

Fig. 2. ROC curves of the informed decoder, the MAP blind decoder and the symmetric Tardos decoder for 3 different code lengths and the wca strategy with $c = 6$ and $n = 1000$ users.

a minimization algorithm described in [16]. This minimization gives a stationary attack channel as introduced in section IV-A.

In [3], the efficiency of the code in term of error bound is proved over all random choices of the code, *i.e.,* for all random choices of the secret vector $\mathbf{p}$, the dictionary of users $\mathbf{X}$ and the strategy $\mathbf{G}$. In our experiment, for one realization of the Monte Carlo, all these variables are randomly sampled. For one realization we use $n = 10^3$ users, with c fixed before running the experiment. We then compute the scores with three decoders. At each realization we store 2×3 scores, as we keep only the biggest score of the colluders and the biggest score of the innocents. Each run encompasses 10^4 realizations.

The false positive is related to $\sigma_{x_{inn}}$, the vector of all innocent-user scores as

$$pfa = \mathbb{P}(max(\sigma_{x_{inn}}) \geq \tau).$$

Hence a false alarm event occurs when at least one innocent user is accused, *i.e.,* one innocent-user score is above the considered threshold τ.

The false negative is related to $\sigma_{x_{coll}}$, the vector of all colluder scores, as

$$pfn = \mathbb{P}(max(\sigma_{x_{coll}}) < \tau).$$

The false negative event occurs when all colluders are missed, *i.e.,* when all colluder scores are below the considered threshold τ.

A. Stability over different collusion channels

Figure 1 shows the ROC curves for the symmetric Tardos decoder, the informed decoder and our MAP blind decoder. We consider a fingerprinting code of length $m = 300$, a maximum number of colluders $c_{max} = 10$ and a true number of colluders $c = 6$. The legend of Figure 1 is set as follows. The uppercase letters "T", "I", "M" are used in this order for the **T**ardos decoder, the **I**nformed decoder and the **M**AP blind decoder. The lowercase letters "r","fc","wca","m" states for **r**andom, **f**lip **c**oin, **w**orst-**c**ase **a**ttack and **m**inority strategies.

For all strategies our decoder leads to enhanced performance as compared to Tardos decoder. Not surprisingly, the informed decoder gives the best performance for all strategies. The stability of the Tardos decoder performance is shown for this set of collusion strategies: unlike the informed and MAP blind decoders, its performance does not vary a lot with the considered strategy. For some strategies, this stability is however achieved at the expense of a loss of performance with respect to the informed and the MAP blind decoders. In particular, the largest gap between the performance of Tardos and informed/MAP blind decoders is reached for the minimum strategy which appears to be the more damaging for Tardos approach [13]. The wca strategy is the worst strategy against the Informed decoder. It is also the worst strategy against our MAP blind decoder for this set of strategies. However it is important to mention that the wca strategy has not been proved to be the worst attack against the MAP blind decoder.

Even if the MAP blind decoder is not quite as good as the informed decoder for the flip coin strategies, its performance is very closed to the informed decoder, the ROC curves are almost overlapped in this case. This is consistent with the coin flip strategy, because only the true coalition size is unknown for the MAP blind decoder. Notice that the performance of the MAP blind decoder against the flip coin attack is a little bit better than the performance of the informed decoder for the random strategy. Nevertheless, these last three considered configurations lead almost to the same performance. It is also the case for the worst case attack against the informed decoder and for the random attack against the MAP Blind decoder.

B. Evaluation over different code lengths

Figure 2 presents the ROC curves for the symmetric Tardos decoder, the informed decoder and our MAP blind decoder

for different code lengths. We evaluate the performance of the decoders for the wca attack since it is the worst attack against our MAP blind decoder among the considered strategies. We use the same set of parameters as in Figure 1 with the same legend terminology. The probability of false alarm and the probability of false negative are set in logarithmic scale in Figure 2.

For all lengths, our decoder results in less decoding errors compared to the symmetric Tardos decoder. In particular, our decoder performance is closer to the informed decoder performance than Tardos decoder. Moreover, the gap between the performance of the Tardos and the MAP blind decoders increases as the length of the code increases.

VI. Conclusion

Our blind Maximum A Posteriori approach works for any probabilistic codes under the marking assumption with acceptable complexity. Promising results compared to the symmetric Tardos decoder are presented. The preliminary results presented here open however some important questions:

1) What is the behaviour of our decoder if the true coalition size is above the maximum coalition size set to the decoder?
2) Is our decoder better than estimation-based decoders, such as in [13], against time varying attacks and stationary attacks?
3) How is linked our decoder functions with the Tardos ones? Some preliminary numerical results, not presented here, gives some correlations in particular asymptotic cases.

These three last issues will be addressed in our future research as the study of the setting of a proper threshold so as to bound false alarm probability.

Acknowledgment

The authors thank Teddy Furon for his valuable help in writing this paper.

References

[1] N. Wagner, "Fingerprinting," in Symposium on Security and Privacy, IEEE Computer Society, 1983, pp. pp. 18–22.
[2] G. R. Blakley, C. Meadows, and G. B. Purdy, "Fingerprinting long forgiving messages," in Advances in Cryptology CRYPTO 85 Proceedings, 1985, vol. 218, pp. 180–189.
[3] G. Tardos, "Optimal probabilistic fingerprint codes," J. ACM, vol. 55, no. 2, p. 10:110:24, 2008.
[4] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," IEEE Transactions on Information Theory, vol. 44, pp. 1897–1905, 1998.
[5] C. Peikert, A. Shelat, and A. Smith, "Lower bounds for collusion-secure fingerprinting," in Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, 2003, pp. 472–479.
[6] O. Blayer and T. Tassa, "Improved versions of tardos fingerprinting scheme," Designs, Codes and Cryptography, vol. 48, pp. 79–103, 2008.
[7] B. Skoric, T. U. Vladimirova, M. Celik, and J. C. Talstra, "Tardos fingerprinting is better than we thought," IEEE Transactions on Information Theory, vol. 54, pp. 3663–3676, 2008.
[8] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of tardos's collusion-secure fingerprinting codes with very short lengths," in Proceedings of the 17th international conference on Applied algebra, algebraic algorithms and error-correcting codes, ser. AAECC'07, 2007, p. 8089.
[9] M. Kuribayashi, N. Akashi, and M. Morii, "On the systematic generation of tardos's fingerprinting codes," in 2008 IEEE 10th Workshop on Multimedia Signal Processing, 2008, pp. 748 –753.
[10] P. Moulin, "Universal fingerprinting: Capacity and random-coding exponents," in IEEE International Symposium on Information Theory, 2008. ISIT 2008, 2008, pp. 220 –224.
[11] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, ser. SODA '09, 2009, p. 336345.
[12] P. Meerwald and T. Furon, "Towards joint tardos decoding: The Don quixote algorithm," in Information Hiding. Springer Berlin Heidelberg, 2011, vol. 6958, pp. 28–42.
[13] T. Furon and L. Perez-Freire, "EM decoding of tardos traitor tracing codes," in Proceedings of the 11th ACM workshop on Multimedia and security, 2009, pp. 99–106.
[14] B. Skoric, S. Katzenbeisser, and M. U. Celik, "Symmetric tardos fingerprinting codes for arbitrary alphabet sizes," Designs, Codes and Cryptography, vol. 46, p. 137166, 2008.
[15] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in ICC '95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on Communications, 1995, vol. 2, 1995, pp. 1009–1013.
[16] T. Furon and L. Perez-Freire, "Worst case attacks against binary probabilistic traitor tracing codes," in Proceedings of First IEEE International Workshop on Information Forensics and Security, 2009, pp. 46–50.