Towards practical joint decoding of binary Tardos fingerprinting codes

Peter Meerwald, Teddy Furon

Toward Practical Joint Decoding of Binary Tardos Fingerprinting Codes

Peter Meerwald and Teddy Furon

***Abstract*—The class of joint decoder in fingerprinting codes is of utmost importance in theoretical papers to establish the concept of fingerprint capacity. However, no implementation supporting a large user base is known to date. This paper presents an iterative decoder which is the first attempt toward practical large-scale joint decoding. The discriminative feature of the scores benefits on one hand from the side-information of previously found users, and on the other hand, from recently introduced universal linear decoders for compound channels. Neither the code construction nor the decoder makes assumptions about the collusion size and strategy, provided it is a memoryless and fair attack. The extension to incorporate soft outputs from the watermarking layer is straightforward. An extensive experimental work benchmarks the very good performance and offers a clear comparison with previous state-of-the-art decoders.**

***Index Terms*—Compound channel, fingerprinting, Tardos codes, traitor tracing.**

I. Introduction

TRAITOR tracing or active fingerprinting has witnessed a flurry of research efforts since the invention of the now well-celebrated Tardos codes [6]. The codes of G. Tardos are order-optimal in the sense that the code length m necessary to fulfill the following requirements (n users, c colluders, probability of accusing at least one innocent below P_{fp}) has the minimum scaling in $\Omega(c^2 \log n P_{\mathsf{fp}}^{-1})$.

A first group of papers analyzes such probabilistic fingerprinting codes from the viewpoint of information theory. They define the worst case attack a collusion of size c can produce, and also the best defense. The main achievement is a saddle point theorem in the game between the colluders and the code designer which establishes the concept of fingerprinting capacity $C(c)$ [1]–[4]. Roughly speaking, for a collusion of maximum size c, the maximum number of users exponentially grows with m with an exponent equal to $C(c)$ to guarantee vanishing probabilities of error asymptotically as the code length increases.

Manuscript received November 22, 2011; revised March 22, 2012; accepted April 09, 2012. Date of publication April 20, 2012; date of current version July 09, 2012. This work was supported in part by the French ANR Project Medievals. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Adnan M. Alattar.

P. Meerwald is with BCT Electronic, A-5020 Salzburg, Austria (e-mail: pmeerw@pmeerw.net).

T. Furon is with INRIA, 35042 Rennes, France (e-mail: teddy.furon@inria.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2012.2195655

Our point of view is much more practical and signal processing oriented. In traitor tracing applications distinct codewords of m bits have been hidden in distributed copies with an appropriate watermarking technique. It implies that we have no choice on m as it depends on the content size and the watermarking embedding rate. Each embedded codeword links a copy of the content to a particular user. The total number of users may not be known in advance like, for instance, in a Video-on-Demand application where clients buy content sequentially. However, at the time a pirated version is discovered, we know that the content has been distributed to n users so far. Our goal is to identify some colluders under the strict requirement that the probability of accusing innocents is below P_{fp}. It is clear that we are not in an asymptotic setup since m and n are fixed. The encoder and the decoder are not informed of the collusion size and its attack, therefore there is no clue whether the actual rate $R = m^{-1} \log_2 n$ is indeed below capacity $C(c)$. After reviewing the construction of Tardos codes and the collusion attack, Section II summarizes important elements of information theory as guidelines for the design of our decoder. It motivates the use of joint decoding, which computes a score per subset of users, as opposed to a single decoder computing a score per user.

A second group of related research works deals with decoding algorithms. As far as a practical implementation of joint decoding is concerned, the literature is very scarce. Amiri proposes a pair decoder tractable on a short list of suspects [7, sec. 5.3]. The idea is proven to be theoretically well grounded against two colluders. However, the sorting is in terms of Hamming distance from the pirated sequence, which seems to be quite an ad hoc choice; no experimental work is conducted. Nuida also proposes a provably secure joint decoder against three colluders whose runtime is longer than one hour for a very small setup ($n = 1000,\ m = 180$) [8].

For a single as well as a joint decoder, a primary challenge is to compute scores that are as discriminative as possible. The earliest decoders proposed in the literature are single decoders not adapting the score computation to the collusion strategy [6], [9]. They rely on an invariance property where, whatever the collusion process under the marking assumption, the statistics of scores of the innocents and the colluders are almost fixed and sufficiently discriminative if the code is long enough. Being inspired by a recent paper on compound channel theory [5], we first propose a generalized linear single decoder which is more discriminative but at the cost of higher complexity. A second approach in the literature aims at first identifying the collusion process, and then at computing more discriminative scores for this specific attack [10]. However, the identifiability

of the attack is a crucial issue when the number of colluders is not known. Again, the application of the concepts from [5] allows us to get inferences about the attack sufficient for deriving highly discriminative scores while avoiding a complete identification of the attack channel. Section III sums up these two families of single decoders and the way we have improved them.

A further difficulty in traitor tracing schemes is the thresholding of the scores to reliably accuse users who are part of the collusion. The value of the threshold is easily set when the statistics of the scores are known, which is the case when the above-mentioned invariance property holds. However, for a general scoring function, these statistics depend on the collusion process which is not known. Section III-C presents a simple idea: there are plenty of codewords which have not been distributed to users. Therefore, it is possible to use them as instantiations of the codeword of an innocent. We propose to estimate the threshold yielding the required probability of false alarm with a rare event estimator.

Section IV focuses on the architecture of our joint decoder based on three primitives: channel inference, score computation, and thresholding. Its iterative nature stems from two ideas. First, the codeword of a newly accused user is integrated as a side information for the next iterations and enables more discriminative score computation. This idea was already implemented for fingerprinting codes based on error correcting codes [11]. We present a way to implement it for Tardos codes by conditioning the probabilities used in the score function. The second idea is joint decoding based on the channel inference.

A last difficulty is to have a fast implementation of the accusation algorithm when facing a large-scale user set. A main advantage of some fingerprinting schemes based on error-correcting codes is to offer an accusation procedure with runtime polynomial in m [11], [12]. In comparison, the well-known Tardos-Škoric single decoder is based on an exhaustive search which has complexity $\Omega(m \cdot n)$ [6], [9]. Since in theory n can asymptotically be in the order of 2^{mR}, decoding of Tardos codes might be intractable. Again, we do not consider such a theoretical setup, but we pay attention to maintain an affordable decoding complexity for orders of magnitude met in practical applications. Compared to prior art of joint decoding [7], [8], our algorithm considers user subsets of bigger size, manages large scale setups, and is faster. Its iterative nature maintains a tractable complexity because users that are unlikely to be guilty are pruned out at each step.

Section V presents our experimental investigations. The first part relies on the marking assumption and compares code lengths with [13]. This reference reaches very small lengths thanks to a particular choice of Gauss-Legendre distribution, but assuming the collusion size is known at the Tardos code construction. It is interesting to see that our decoder obtains competitive lengths while keeping the original code construction. The second part uses the soft outputs of a watermarking decoder as tested with the Tardos code in [14] and with the error correction code (ECC)-based fingerprinting code in [15] and [16]. The number of users ranges from 10^4 to more than 10^7. This latter impressive setup comes from [16] where the authors manage such a large number of users thanks to list decoding of ECC-based fingerprinting. As far as we know, our paper presents for the first time experimental results on such a large scale for the Tardos code. Soft watermark decoding achieves tangible performance enhancements contrary to the conclusions drawn in [14]. Overall, the comparisons to related works with their exact setup show the benefits of our decoder: better decoding performance with a controlled probability of false alarm and an acceptable runtime.

II. Tardos Code and Collusion Model

We briefly review the construction and some well-known facts about Tardos codes.

A. Construction

The binary code is composed of n codewords of m bits. The codeword $\mathbf{x}_j = (x_j(1), \ldots, x_j(m))^T$ identifying user $j \in \mathcal{U} = [n]$, where $[n] := \{1, \ldots, n\}$, is composed of m binary symbols independently drawn at the code construction s.t. $\mathbb{P}(x_j(i) = 1) = p_i$, $\forall i \in [m]$. At initialization, the auxiliary variables $\{p_i\}_{i=1}^m$ are independent and identically drawn according to distribution $f(p) : [0,1] \longrightarrow \mathbb{R}^+$. This distribution is a parameter which is public. Tardos originally proposed in [6] $f_T(p) \propto 1/\sqrt{p(1-p)}$ for $p \in [t, 1-t]$ where $t \ll 1$ is the cutoff parameter. Both the code $\Xi = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and the auxiliary sequence $\mathbf{p} = (p_1, \ldots, p_m)^T$ must be kept as secret parameters.

B. Collusion Attack Over Code Symbols

The collusion attack or collusion channel describes the way the c colluders $\mathcal{C} = \{j_1, \ldots, j_c\}$ merge their binary codewords $\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_c}$ to forge the binary pirated sequence $\mathbf{y}$. We restrict our attention to a memoryless multiple access channel, which is fair in the sense that all colluders participate equally in the forgery. This assumption is widely used, and justified theoretically [1, sec. 3.2] (in terms of capacity, i.e., asymptotically with the code length) when the secret key is only shared between encoder and decoder: The colluders know neither the codeword of any other user, nor the distribution of the codewords. This is the case for a Tardos code because $\mathbf{p}$ is secret. Moreover, identifying all the colluders is hopeless (c.f. detect-all scenario, see Section II-D) if the attack is not fair because some colluders might be almost idle [1, lemma 3.2].

This leads to a $2 \times (c+1)$ probability transition matrix $[\mathbb{P}(Y|\Phi)]$ where $\Phi = \sum_{j \in \mathcal{C}} X_j$ is the random variable counting the number of "1" symbols the colluders received out of c symbols. A common parameter of the collusion attack on binary codes is denoted by the vector $\boldsymbol{\theta}_c = (\theta_c(0), \ldots, \theta_c(c))^T$ with $\theta_c(\varphi) = \mathbb{P}(Y = 1 | \Phi = \varphi)$. The usual working assumption, so-called *marking assumption* [17], imposes that $\theta_c(0) = 1 - \theta_c(c) = 0$. The set of collusion attacks that c colluders can lead under the marking assumption is denoted by Θ_c

$$\Theta_c = \{\boldsymbol{\theta} \in [0,1]^{c+1},\ \theta(0) = 1 - \theta(c) = 0\}. \tag{1}$$

Examples of attacks following this model are given, for instance, in [10]. The remainder of the paper assumes this collusion model, except for the simulations on real samples over an additive white Gaussian noise (AWGN) channel which are based on the extension presented as follows.

C. Collusion Attack Over Real Samples

The marking assumption is an unrealistic restriction for traitor tracing with multimedia content as the colluders are not limited to the copy-and-paste strategy for each symbol as described in the previous section. They can merge the samples of their content versions (audio samples, pixels, DCT coefficients, etc.) in addition to traditional attempts to compromise the watermark. This may result in erroneously decoded symbols or erasures from the watermarking layer. Relaxing the marking assumption leads to two approaches. In the combined digit model assumed in [18] and [19], the watermark decoder is indeed composed of multiple binary detectors, one per symbol of the alphabets: for the binary alphabet, both symbols may be detected in case of a merge. In [14] and [15], the watermark decoder has a single but scalar output y'. In brief, this soft decision is clearly negative (positive) if symbol "0" (respectively "1") is detected, and around 0 in case of a merge. This section extends the model of collusion to this latter approach, replacing the probability transition $2 \times (c+1)$ matrix $[\mathbb{P}(Y|\Phi)]$ by $c+1$ probability density functions $\{\theta_c(y'|\varphi)\}_{\varphi=0}^{c}$.

It is challenging if not impossible to exhibit a model encompassing all the merging attacks while being relevant for a majority of watermarking techniques. Our approach is pragmatic. The sequence $\mathbf{y}' \in \mathbb{R}^m$ is extracted from the pirated copy such that $y'(i) = 2y(i)-1$ if the signal is perfectly watermarked with binary symbol $y(i)$. To reflect the merging attack, the colluders forge values $z(i) \in [-1,1]$ and add noise: $y'(i) = z(i) + n(i)$ with $n(i) \sim \mathcal{N}(0, \sigma_{\mathsf{n}}^2)$. This would be the case, as sketched in the left part of Fig. 3, for a spread spectrum watermarking where a symbol is embedded per block of content with an antipodal [a.k.a. binary phase shift keying (BPSK)] modulation of a secret carrier [14], [16].

The colluders have two strategies to agree on $\mathbf{z}$. In the first strategy, they collude according to the marking assumption (i.e., they copy-and-paste one of their blocks of samples) and add noise: $\mathbf{z} \in \{-1,1\}^m$ and the probability that $z=1$ is given by the components of $\boldsymbol{\theta}_c$. This gives the following pdf:

$$\theta_c^{(II)}(y'|\varphi) = \frac{\left(\theta_c(\varphi)e^{-(y'-1)^2/2\sigma_{\mathsf{n}}^2} + (1-\theta_c(\varphi))e^{(y'+1)^2/2\sigma_{\mathsf{n}}^2}\right)}{\sqrt{2\pi\sigma_{\mathsf{n}}^2}}. \quad (2)$$

Except for $\varphi \in \{0, c\}$, the pdfs have *a priori* two modes (hence the superscript II). This model is parameterized by $(\boldsymbol{\theta}, \sigma_{\mathsf{n}}^2)$.

In the second strategy, the colluders select $z(i) = \mu(\varphi(i)) \in [-1,1]$ s.t. the pdf is as follows:

$$\theta_c^{(I)}(y'|\varphi) = \frac{e^{-(y'-\mu(\varphi))^2/2\sigma_{\mathsf{n}}^2}}{\sqrt{2\pi\sigma_{\mathsf{n}}^2}}. \quad (3)$$

An equivalent of the marking assumption would impose that $\mu(0) = -1$ and $\mu(c) = 1$. The pdfs have a unique mode (hence the superscript I). This model is parameterized by $(\boldsymbol{\mu}, \sigma_{\mathsf{n}}^2)$. Fig. 1 gives some examples of such pdfs.

We use these extended models for collusion inference in Section V-B to show how our algorithm can handle the soft outputs of a watermarking decoder.

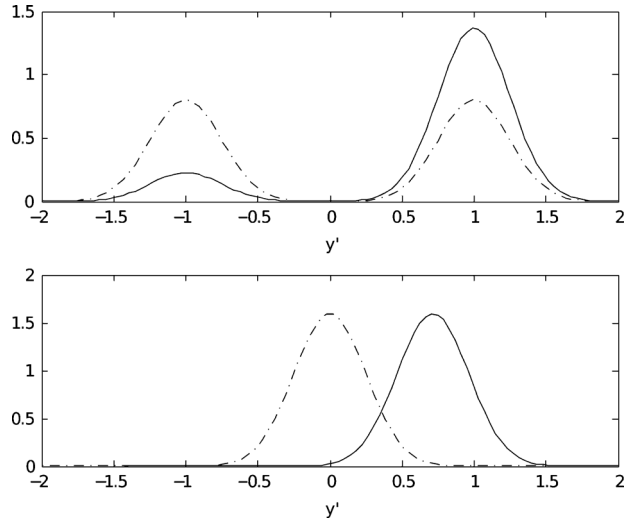

Fig. 1. Examples of pdf $\theta_7(y'|6)$ for the models with $\sigma_{\mathsf{n}}^2 = 0.25$: (top) two modes [$II$; see (2)] with (solid) interleaving attack ($\theta(\varphi) = \varphi/c$) and (dashed) the coin-flip attack ($\theta(\varphi) = 1/2$ for $0 < \varphi < c$). (bottom) One mode [I; see (3)] with (solid) averaging attack ($\mu(\varphi) = 2c^{-1}\varphi - 1$) and (dashed) set to 0 attack ($\mu(\varphi) = 0$ for $0 < \varphi < c$).

D. Accusation

Denote $\mathcal{A} \subset \mathcal{U}$ the set of users accused by the decoder. The probability of false positive is defined by $P_{\mathsf{fp}} = \mathbb{P}(\mathcal{A} \not\subset \mathcal{C})$. In practice, a major requirement is to control this feature so that it is lower than a given significance level.

In a detect-one scenario, $\mathcal{A}$ is either a singleton or the empty set. A good decoder has a low probability of false negative defined by $P_{\mathsf{fn}} = \mathbb{P}(\mathcal{A} = \emptyset)$. In a detect-many scenario, several users are accused, and a possible figure of merit is the number of caught colluders: $|\mathcal{A} \cap \mathcal{C}|$. In the literature, there exists a third scenario, so-called detect-all, where a false negative happens if at least one colluder is missed. This paper only considers the first two scenarios.

E. Guidelines From Information Theory

This paper does not pretend to make any new theoretical contribution, but presents some recent elements to stress guidelines when designing our practical decoder.

A *single decoder* computes a score per user. It accuses users whose score is above a threshold (detect-many scenario) or the user with the biggest score above the threshold (detect-one scenario). Under both scenarios and provided that the collusion is fair, the performance of such decoders is theoretically bounded by the achievable rate $R_S(f, \boldsymbol{\theta}_c) = I(X;Y|P,\boldsymbol{\theta}_c) = \mathbb{E}_{P\sim f}[I(X;Y|p,\boldsymbol{\theta}_c)]$. A fundamental result is that, for a given collusion size c, there exists an equilibrium $(\breve{f}_{c,S}, \breve{\boldsymbol{\theta}}_{c,S})$ to the max-min game between the colluders (who select $\boldsymbol{\theta}$) and the code designer (who selects f) as defined by $\max_f \min_{\boldsymbol{\theta}\in\Theta_c} R_S(f,\boldsymbol{\theta})$ in [22, th. 4].

A *joint decoder* computes a score per subset of $\ell \leq c$ users and accuses the users belonging to subsets whose score is above a threshold or only the most likely guilty amongst these users. Under both scenarios and provided that the collusion is fair, the performance of such decoders is theoretically bounded by the achievable rate

$R_J(f,\boldsymbol{\theta}_c) = \ell^{-1} I(\Phi;Y|P,\boldsymbol{\theta}_c) = \ell^{-1}\mathbb{E}_{P\sim f}[I(\Phi;Y|p,\boldsymbol{\theta}_c)]$. The random variable Φ denotes the sum of the subset user symbols. Moreover, for a given collusion size c, there also exists an equilibrium $(\breve{f}_{c,J}, \breve{\boldsymbol{\theta}}_{c,J})$ to the max-min game $\max_f \min_{\boldsymbol{\theta}\in\Theta_c} R_J(f,\boldsymbol{\theta})$ [2, th. 4].

Yet, the code designer needs to bet on a collusion size c' in order to use the optimal distribution $\breve{f}_{c',S}$ (or $\breve{f}_{c',J}$ if the decoder is joint).

Asymptotically, as $c \to +\infty$, both $\min_{\boldsymbol{\theta}} R_J(f_T,\boldsymbol{\theta})$ and $\min_{\boldsymbol{\theta}} R_S(f_T,\boldsymbol{\theta})$ quickly approach the equilibrium value of the respective max-min game [2, Fig. 2]. Huang and Moulin proved $\breve{f}_{c,J}$ converges to f_T, the distribution originally proposed by Tardos [2, cor. 7].

Despite the division by ℓ in the expression of $R_J(f,\boldsymbol{\theta})$, it appears that $R_S(f,\boldsymbol{\theta}) \leq R_J(f,\boldsymbol{\theta})$, $\forall\boldsymbol{\theta}$ [1, eq. (3.4)]. This tells us that a joint decoder is theoretically more powerful than a single decoder. However, a joint decoder needs to compute $\Omega(n^\ell)$ scores since there are $\binom{n}{\ell}$ subsets of size ℓ. This complexity is absolutely intractable for large-scale applications even for a small ℓ. This explains why, so far, joint decoders were only considered theoretically to derive fingerprinting capacity. Our idea is that there is no need to consider all these subsets since a vast majority is only composed of innocent users. Our decoder iteratively prunes out users deemed as innocents and considers the subsets over the small set of remaining suspects.

This iterative strategy results in a decoder which is a mix of single and joint decoding. Unfortunately, it prevents us from taking advantage of the game theory theorems mentioned previously. We cannot find the optimal distribution f and the worst collusion attack against our decoder. Nevertheless, our decoder works with any distribution f under the conditions stated in Section III. For all these reasons, the experiments of Section IV are done with the most common Tardos distribution f_T.

Fernandez and Soriano proposed an iterative accusation process of an error correcting code based fingerprinting scheme [11]. Each iteration takes advantage of the codewords of colluders already identified in the previous iterations. The same idea is possible with Tardos fingerprinting code. This is justified by the fact that the side information Δ, defined as the random variable sum of the already identified colluder symbols, increases the mutual information: $I(\Phi;Y|P,\boldsymbol{\theta}_c) \leq I(\Phi;Y|P,\boldsymbol{\theta}_c,\Delta)$. Indeed, side information helps more than joint decoding as proved by [1, eq. (3.3)].

The guidelines can be summarized as follows: use the continuous Tardos distribution f_T for code construction, integrate the codewords of already identified colluders as side information, and finally use a joint decoder on a short list of suspects.

III. Single Decoder Based on Compound Channel Theory and Rare Event Analysis

This section first reviews some single decoders and presents new decoders based on compound channel theory and rare event analysis. The first difficulty is to compute a score per user such that the colluders are statistically well separated from the innocents scores. The second difficulty is to set a practical threshold such that the probability of false positive is under control.

Detection theory tells us that the score given by the log-likelihood ratio (LLR)

$$s_j = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i),\boldsymbol{\theta}_c)}{\mathbb{P}(y(i)|\boldsymbol{\theta}_c)} \tag{4}$$

is optimally discriminative in the Neyman-Pearson sense to decide the guiltiness of user j. For the sake of a lighter expression, we omit the dependence of the involved probabilities on $p(i)$ (see their computation in Section IV-B). Yet, the LLR needs the knowledge of the true collusion attack $\boldsymbol{\theta}_c$ which prevents the use of this optimal single decoder in practical settings. Some papers proposed a so-called "Learn and Match" strategy using the LLR score tuned on an estimation $\hat{\boldsymbol{\theta}}$ of the attack channel [10]. Unfortunately, a lack of identifiability obstructs a direct estimation from $(\mathbf{y},\mathbf{p})$ (see Section III-B). Indeed, the estimation is sound only if c is known, and if the number of different values taken by p is bigger than[1] or equal to $c-1$. This is because $\mathbb{P}(Y=1|\boldsymbol{\theta},p)$ is a polynomial in p of degree at most c (see (16) with $u=0$ and $v=0$) going from point $(0,0)$ to $(1,1)$, and we need $c-1$ more points to uniquely identify this polynomial. To overcome this lack of information about c, an expectation-maximization (E.-M.) approach has been proposed in [10]. Yet, it is not satisfactory since it does not scale well with the number of users. Moreover, the setting of the threshold was not addressed.

On the other hand, there are decoders that do not adapt their score computation to the collusion. This is the case of the score computation originally proposed by Tardos [6], and later on improved by Škoric *et al.* [9]. It relies on an invariance property: the statistics of the scores, up to the second order, do not depend on the collusion attack channel $\boldsymbol{\theta}$, but only on the collusion size c [20]. Thanks to this invariance w.r.t. the collusion attack, there exists a threshold τ guaranteeing a probability of false positive below P_{fp} while keeping the false negative away from 1 provided that the code is long enough, i.e., $m = \Omega(c^2 \log n P_{\text{fp}}^{-1})$. However, there is a price to pay: the scores are clearly less discriminative than the LLR.

Some theoretical papers [21, sec. V], [1, sec. 5.2] promote another criterion, so-called "universality", for the design of decoders. The performance (usually evaluated as the achievable rate or the error exponent) when facing a collusion channel $\boldsymbol{\theta}_c$ should not be lower than the performance against the worst attack $\boldsymbol{\theta}_c^\star$. In a sense, it sends a warning to the "Learn and Match" strategy. Suppose that $\boldsymbol{\theta}_c \neq \boldsymbol{\theta}_c^\star$ and that, for some reasons, the estimation of the collusion attack is of poor quality. In any case, a mismatch between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}_c$ should not ruin the performance of the decoder to the point it is even lower than what is achievable under the worst attack $\boldsymbol{\theta}_c^\star$. The previously cited [1], [21] recommend the single universal decoder based on the empirical mutual information $I(\mathbf{x};\mathbf{y}|\mathbf{p})$ (or empirical equivocation for joint decoder). The setting of the threshold depends on the desired error exponent of the false positive rate. Therefore, it is valid only asymptotically.

To summarize, there have been two approaches: adaptation or nonadaptation to the collusion process. The first class is not very well grounded since the identifiability of the collusion attack is an issue and the impact of a mismatch has not been studied. The

[1]This is the case in this paper since we opt for the continuous Tardos distribution f_T.

second approach is more reliable, but with a loss of discrimination power compared to the optimal LLR. The next sections present two new decoders belonging to both approaches based on the compound channel theory.

A. Some Elements on Compound Channels

Recently, in the setup of digital communication through compound channels, Abbe and Zheng [5] proposed universal decoders which are linear, i.e., in essence very simple. This section summarizes this theory and the next one proposes two applications for Tardos single decoders. A compound channel Ψ is a set of channels, here discrete memoryless channels $X \in \mathcal{X} \rightarrow Y \in \mathcal{Y}$ defined by their probability transition matrix $W_\psi = [\mathbb{P}(Y|X,\psi)]$ parameterized by $\psi \in \Psi$. The encoder shares a code book $\Xi = \{\mathbf{x}_j\}_{j=1}^n \in \mathcal{X}^{m\times n}$ with the decoder. Its construction is assumed to be a random code realization from a provably good mass distribution P_X. After receiving a channel output $\mathbf{y} \in \mathcal{Y}^m$, the decoder computes a score per codeword $\mathbf{x}_j$, $j \in [n]$ and yields the message associated with the codeword with the biggest score. The decoder is linear if the score has the following structure:

$$s_j = \sum_{i=1}^{m} d(x_j(i), y(i)) \tag{5}$$

with $d(\cdot,\cdot) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. For instance, score (4), so-called MAP decoder in digital communications [5], is linear with $d(x,y) = \log(\mathbb{P}(y|x,\psi)/\mathbb{P}(y|\psi))$. However, in the compound channel setup, the decoder does not know through which channel of Ψ the codeword has been transmitted, and therefore it cannot rely on the MAP.

We are especially interested in two results. First, if Ψ is *one-sided* with respect to the input distribution (see Definition 1 as follows), then the MAP decoder tuned on the worst channel $W_{\psi^\star}$ is a linear universal decoder [5, lemma 5]. If $\Psi = \bigcup_{k=1}^K \Psi_k$ with K finite and Ψ_k one-sided w.r.t. the input distribution $\forall k \in [K]$, then the following *generalized* linear decoder is universal [5, th. 1] where the score of a codeword is the maximum of the K MAP scores tuned on the worst channel $W_{\psi_k^\star}$ of each Ψ_k

$$s_j = \max_{k\in[K]} \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i),\psi_k^\star)}{\mathbb{P}(y(i)|\psi_k^\star)}. \tag{6}$$

Definition 1 (One-Sided Set of [5, Def. 3]): A set Ψ is one-sided with respect to an input distribution P_X if:

1) the following minimizer is unique:

$$\psi^\star = \arg\min_{\psi\in\mathsf{cl}(\Psi)} \mathcal{I}(P_X,\psi) \tag{7}$$

with $\mathcal{I}(P_X,\psi)$ the mutual information $I(X;Y)$ with $(X,Y) \sim P_X \circ W_\psi$ (where $P\circ W$ denotes the joint distribution with P the distribution of X and W the conditional distribution), and $\mathsf{cl}(\Psi)$ the closure of Ψ;

2) $\forall \psi \in \Psi$

$$D(P_X\circ W_\psi \| P_X \times P_{Y,\psi^\star}) \geq D(P_X\circ W_\psi \| P_X \circ W_{\psi^\star}) + D(P_X \circ W_{\psi^\star} \| P_X \times P_{Y,\psi^\star}) \tag{8}$$

with $D(\cdot\|\cdot)$ the Kullback-Leibler distance, $P_{Y,\psi}$ the marginal of Y induced by $P_X \circ W_\psi$, and $P_X \times P_{Y,\psi}$ the product of the marginals.

B. Application to Single Tardos Decoders

Contrary to the code construction phase, it is less critical at the decoding side to presume that the real collusion size c is less than or equal to a given parameter c_{max}. This parameter can be set to the largest number of colluders the fingerprinting code can handle with a reasonable error probability knowing (m,n). Another argument is that this assumption is not definitive. If the decoding fails because the assumption does not hold true, nothing prevents us from relaunching decoding with a bigger c_{max}. Let us assume $c \leq c_{\mathsf{max}}$ in the sequel.

1) Nonadaption to Collusion Process: A first guideline inspired from the work [5] is straightforward: The collusion channel belongs to the set $\bigcup_{k=2}^{c_{\mathsf{max}}} \Theta_k$ as defined in (1), and thanks to [5, lemma 4] each convex set Θ_k is one-sided w.r.t. any distribution f. According to [5, th. 1], the decoder should then be a generalized linear decoder

$$s_j = \max_{k\in[2,\ldots,c_{\mathsf{max}}]} \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i),\boldsymbol{\theta}_{k,f}^\star)}{\mathbb{P}(y(i)|\boldsymbol{\theta}_{k,f}^\star)} \tag{9}$$

where $\boldsymbol{\theta}_{k,f}^\star = \arg\min_{\Theta_k} R_S(f,\boldsymbol{\theta}),\ \forall k \in [2,\ldots,c_{\mathsf{max}}]$. This decoder does not adapt its score computation to the collusion attack.

2) Adaption to Collusion Process: The second idea inspired from the work [5] is more involved as the lack of identifiability turns to our advantage. The true collusion channel $\boldsymbol{\theta}_c$ has generated data $\mathbf{y}$ distributed as $\mathbb{P}(y|p,\boldsymbol{\theta}_c)$. Let us define the class $\mathcal{E}(\boldsymbol{\theta}_c) = \{\tilde{\boldsymbol{\theta}} | \mathbb{P}(y|p,\tilde{\boldsymbol{\theta}}) = \mathbb{P}(y|p,\boldsymbol{\theta}_c),\ \forall (y,p) \in \{0,1\}\times[0,1]\}$. From [22, prop. 3], we know that $\mathcal{E}(\boldsymbol{\theta}_c)$ is not restricted to the singleton $\{\boldsymbol{\theta}_c\}$ because for any $c' > c$ there exists one $\tilde{\boldsymbol{\theta}}_{c'} \in \mathcal{E}(\boldsymbol{\theta}_c)$. This is true especially for $c' = c_{\mathsf{max}}$. Asymptotically with the code length, the consistent maximum likelihood estimator (MLE) parameterized on c_{max}, as defined in (18), yields an estimation $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} \approx \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}} \in \mathcal{E}(\boldsymbol{\theta}_c)$ with increasing accuracy. This is not an estimation of the true collusion attack because $c \neq c_{\mathsf{max}}$ *a priori*. Therefore, we prefer to refer to $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$ as a collusion inference, and the scoring uses this inference as follows:

$$s_j = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i),\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}{\mathbb{P}(y(i)|\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}. \tag{10}$$

Suppose that the MLE tuned on c_{max} provides a perfect inference $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} = \tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, we then succeed to restrict the compound channel to the discrete set $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$ which we define as the restriction of $\mathcal{E}(\boldsymbol{\theta}_c)$ to collusions of size $\tilde{c} \leq c_{\mathsf{max}}$. Appendix A shows that $\mathcal{E}_{c_{\mathsf{max}}}(\boldsymbol{\theta}_c)$ is one-sided w.r.t. $f = f_T$, and its worst attack is indeed $\tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$. In [5, lemma 5] the authors justify the use of the MAP decoder (4) tuned on $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$. Its application leads to a more efficient decoder since $R_S(f_T,\tilde{\boldsymbol{\theta}}_{c_{\mathsf{max}}}) \geq R_S(f_T,\boldsymbol{\theta}_{c_{\mathsf{max}},f_T}^\star)$. This decoder pertains to the approach based on score adaptation, with noticeable advantages: it is better theoretically grounded and it is far less complex than the iterative E.-M. decoder of [10].

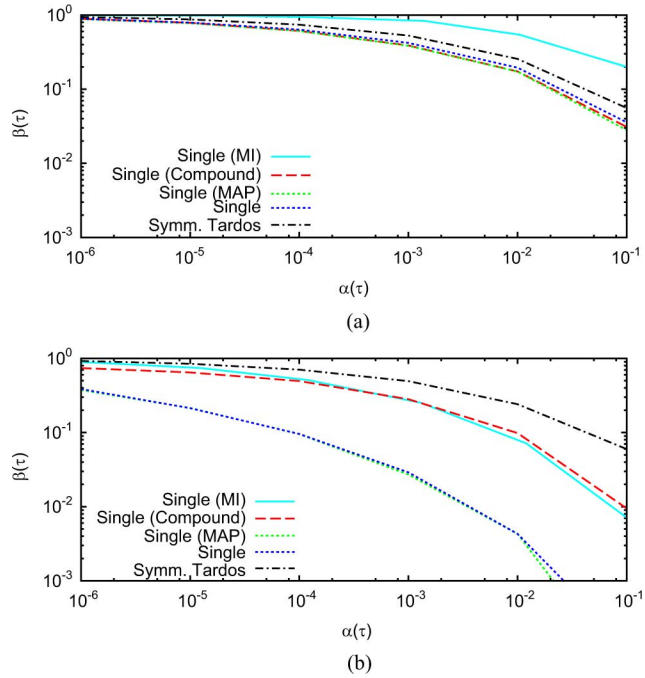

Fig. 2. DET plot for several decoders; $m = 512$, $c = 5$, $c_{\max} = 8$. Single (MI) is the decoder based on empirical mutual information [1], Single (Compound) relates to (9), Single (MAP) is (4), Single is the LLR on $\hat{\boldsymbol{\theta}}_{c_{\max}}$ (10), and Symm. Tardos is the symmetric version of Tardos scores proposed by Škoric *et al.* in [9]: (a) *Worst case* attack and (b) *Majority* attack.

Fig. 2 illustrates the detection error tradeoff (DET) per user for the single decoders discussed so far with $m = 512$ and $c = 5$ colluders performing *worst case* (i.e., minimizing $R_S(f_T, \boldsymbol{\theta})$ over Θ_5) and *majority* attack ($\theta_{5,\text{maj}} = (0, 0, 0, 1, 1, 1)^T$). For this figure, the false positive $\alpha(\tau)$ and the false negative $\beta(\tau)$ probabilities are defined *per user* as follows:

$$\alpha(\tau) = \mathbb{P}(s(\mathbf{x}_{\text{inn}}, \mathbf{y}, \mathbf{p}) > \tau) \tag{11}$$

$$\beta(\tau) = \mathbb{P}(s(\mathbf{x}_{j_1}, \mathbf{y}, \mathbf{p}) \leq \tau) \tag{12}$$

where $\mathbf{x}_{\text{inn}}$ is a random variable denoting the codeword of an innocent user and $\mathbf{x}_{j_1}$, the codeword of the first colluder. The *single* decoder is tuned on the collusion inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ (with $c_{\max} = 8$) and performs almost as well as the MAP decoder having knowledge of $\boldsymbol{\theta}$. The DET of the symmetric Tardos score is invariant w.r.t. the collusion attack. The generalized linear decoder of (9) denoted *compound* takes little advantage of the fact that the majority attack is much milder than the worst attack. For a fair comparison, the single decoder based on the empirical mutual information [1] assumes a Tardos distribution uniformly quantized to 10 bins; better results (yet still below the *single* decoder) can be obtained when tuned to the optimal discrete distribution for $c = 5$ colluders [23].

The similarities between compound channel and fingerprinting has been our main inspiration, however some differences prevent any claim of optimality. First, in the compound channel problem, there is a unique codeword that has been transmitted, whereas in fingerprinting, $\mathbf{y}$ is forged from c codewords like in a multiple access channel. Therefore, the derived single decoders are provably good for tracing a given colluder (detect-one scenario), but they might not be the best when looking for more colluders (detect-many scenario). The second difference is that the decoder should sometimes refuse to accuse any user to reduce the possibility of falsely accusing innocent users. The setting of a threshold is clearly missing for the moment.

C. Setting Threshold With Rare Event Analysis

This section explains how we set a threshold τ in accordance with the required P_{fp} thanks to a rare event analysis. Our approach is very different than [1], [3], [6], and [21] where a theoretical development either finds a general threshold suitable when facing a collusion of size c, or equivalently, where it claims a reliable decision when the rate is below the capacity which depends on c. Simone and Škoric recently made a precise analysis of the pdf of the score of an innocent user [24], but it needs the collusion attack channel $\boldsymbol{\theta}_c$. Neither c nor $\boldsymbol{\theta}_c$ is needed in our threshold estimation, but it only holds for a given couple $(\mathbf{p}, \mathbf{y})$ and a known n. Once these are fixed, the score $s_j = s(\mathbf{x}_j, \mathbf{y}, \mathbf{p})$ is a deterministic function from $\{0, 1\}^m$ to $\mathbb{R}$. Since the codewords of the innocent users are i.i.d. and $c \ll n$, we have

$$\begin{aligned} P_{\text{fp}} &= 1 - (1 - \mathbb{P}(s(\mathbf{x}_{\text{inn}}, \mathbf{y}, \mathbf{p}) > \tau))^{n-c} \\ &\approx n \cdot \mathbb{P}(s(\mathbf{x}_{\text{inn}}, \mathbf{y}, \mathbf{p}) > \tau). \end{aligned} \tag{13}$$

The number of possible codewords can be evaluated as the number of typical sequences, i.e., in the order of $2^{m\mathbb{E}_{P\sim f}[h_b(p)]}$, with $h_b(p)$ the entropy in bits of a Bernoulli random variable $B(p)$. $\mathbb{E}_{P\sim f_T}[h_b(p)] \approx 0.557$ bits, which leads to a far bigger number of typical sequences than n (say $m \geq 300$ and $n \leq 10^8$ in practice). This shows that plenty of codewords have not been created when a pirate copy is found. Therefore, we consider them as occurrences of $\mathbf{x}_{\text{inn}}$ since we are sure that they have not participated in the forgery of $\mathbf{y}$. The idea is then to estimate τ s.t. $\mathbb{P}(s(\mathbf{x}_{\text{inn}}, \mathbf{y}, \mathbf{p}) > \tau) = n^{-1} P_{\text{fp}}$ with, for instance, a Monte Carlo simulation with newly created codewords.

The difficulty lies in the order of magnitude. Some typical requirements are $n \approx 10^6$ and $P_{\text{fp}} = 10^{-4}$, hence an estimation of τ corresponding to a probability as small as $\pi = 10^{-10}$. This is not tractable with a basic Monte Carlo on a regular computer because it requires $O(\pi^{-1})$ runs. However, the new estimator based on rare event analysis proposed in [25] performs remarkably fast within this range of magnitude. It produces $\hat{\tau}$ and a C% confidence interval[2] $[\tau^-, \tau^+]$ with only $O(\log(\pi^{-1}))$ runs. In our decoder, we compare the scores to τ^+ (i.e., a pessimistic estimate of τ) to ensure a total false positive probability lower than P_{fp}. Last but not least, this approach works for any deterministic scoring function $s(\cdot)$ and is also applied to joint decoding in Section IV-C.

IV. Iterative, Joint Decoding Algorithm

This section extends the single decoder based on the collusion inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ toward joint decoding, according to the guidelines of Section II-E. Preliminary results about these key ideas were first presented in [26] and [27]. The description as follows makes references to the lines of the pseudo-code of Algorithm 1.

[2]In practice, we set $C = 95$, i.e., we are 95% sure that the true τ lies in this interval.

A. Architecture

The first principle is to iterate the score computation and include users accused in previous iterations as side-information to build a more discriminative test. Let $\mathcal{U}_{\mathsf{SI}} = \emptyset$ denote the initially empty set of accused users (line 1). In each iteration we aim at identifying a (possibly empty) set of users $\mathcal{A}$ (lines 8 and 20) and then update $\mathcal{U}_{\mathsf{SI}}$ with $\mathcal{A}$ (line 24).

Second, we additionally compute scores for subsets of t users of $\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}$, $t \leq c_{\max}$ (line 13). There are $\binom{|\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}|}{t}$ such subsets. As n is large, enumerating and computing a score for each subset is intractable even for small t. The idea here is to find a restricted set $\mathcal{U}^{(t)} \subseteq \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}$ of $n^{(t)} = |\mathcal{U}^{(t)}|$ users (line 11) that are the most likely to be guilty and to keep $p^{(t)} = \binom{n^{(t)}}{t}$ approximately constant from one iteration to another and within our computation resources. We gradually reduce $n^{(t)}$ by pruning out users who are unlikely to have taken part in the collusion when going from single $(t = 1)$ decoding to pair $(t = 2)$ decoding, etc. If $n^{(t)} = O(n^{1/t})$, then score computation of t-subsets over the restricted user set is within $O(n)$ just like for the single decoder. By abuse of notation, $\binom{\mathcal{U}^{(t)}}{t}$ denotes the set of all t-subsets of $\mathcal{U}^{(t)}$ (line 13).

Initially, the single decoder computes the score of all users (line 6) and accuses those whose score is above the estimated threshold τ^+ (line 8). If this happens, these users are included in $\mathcal{U}_{\mathsf{SI}}$ (line 24), and the single decoder restarts by better estimating the collusion inference and computing more discriminative scores thanks to the side-information. If nobody is accused, we rank the users according to their "single" score, i.e., the top-ranked user is most likely to be a colluder, and only the $n^{(2)}$ first users are included in $\mathcal{U}^{(2)}$. This list of suspects is passed to the joint decoder for $t = 2$, i.e., a pair decoder.

The joint decoder produces a new list of scores computed from subsets of t users (line 13), which—according to theoretical results [1], [3]—are more discriminative as t increases. Yet, the accusation and the pruning operations at this stage are more involved than with the single decoder. Denote $\mathcal{T}^\diamond \subseteq \mathcal{U}^{(t)}$ the t-subset of users with the highest score (line 15). If this score is above the threshold, the joint decoder tries to accuse the most likely colluder within $\mathcal{T}^\diamond$ (lines 17–21; see Section IV-C). Therefore, at most one user is accused at this step (contrary to the single decoder, which may accuse more than one user). If this happens, $\mathcal{U}_{\mathsf{SI}}$ is updated (line 24), and the single decoder restarts taking into account this new side-information. If no accusation can be made by the joint t-subset decoder, we generate a new and shorter list of suspects $\mathcal{U}^{(t+1)}$ based on the ranking of joint scores (line 11) that is fed to the subsequent $(t+1)$ joint decoding stage (see Section IV-C).

In the detect-one scenario, the algorithm stops after the first accusation. We restrict the subset size to $t \leq t_{\max}$, with $t_{\max} = 5$. This is not a severe limitation as for moderately large c, the decoding performance advantage of the joint decoder quickly vanishes [1]. In the detect-many scenario, iteration stops when $|\mathcal{U}_{\mathsf{SI}}| = c_{\max}$ or t reaches $\min(t_{\max}, c_{\max} - |\mathcal{U}_{\mathsf{SI}}|)$ and no further accusation can be made. The set $\mathcal{U}_{\mathsf{SI}}$ then contains the user indices to be accused. Algorithm 1 gives the architecture of the accusation process for the catch-many scenario.

Algorithm 1 Iterative Joint Tardos Decoder

Require: $\mathbf{y}$, Ξ, $\mathbf{p}$, $c_{\max}$, $t_{\max} \leq c_{\max}$, $n^{(t)}$, P_{fp}

1: $\mathcal{U} \leftarrow \{j | 1 \leq j \leq n\}, \mathcal{U}_{\mathsf{SI}} \leftarrow \emptyset$
2: **repeat**
3: $\quad t \leftarrow 1$
4: $\quad \hat{\boldsymbol{\theta}}_{c_{\max}} \leftarrow \texttt{infer}(\mathbf{y}, \mathbf{p}, \mathcal{U}_{\mathsf{SI}}, c_{\max})$
5: $\quad \mathbf{W} \leftarrow \texttt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{\mathsf{SI}})$
6: $\quad \mathbf{s} \leftarrow \texttt{scores}(\mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}, \Xi, \mathbf{W})$
7: $\quad \tau^+ \leftarrow \texttt{threshold}(\mathbf{p}, \mathbf{W}, P_{\mathsf{fp}}, n, t)$
8: $\quad \mathcal{A} \leftarrow \{j \in \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}} | s_j > \tau^+\}$
9: $\quad$ **while** $\mathcal{A} = \emptyset$ **and** $t < \min(t_{\max}, c_{\max} - |\mathcal{U}_{\mathsf{SI}}|)$ **do**
10: $\quad\quad t \leftarrow t + 1$
11: $\quad\quad \mathcal{U}^{(t)} \leftarrow \texttt{top}(\mathbf{s}, \mathcal{U} \setminus \mathcal{U}_{\mathsf{SI}}, n^{(t)})$
12: $\quad\quad \mathbf{W} \leftarrow \texttt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{\mathsf{SI}})$
13: $\quad\quad \mathbf{s} \leftarrow \texttt{scores}(\binom{\mathcal{U}^{(t)}}{t}, \Xi, \mathbf{W})$
14: $\quad\quad \tau^+ \leftarrow \texttt{threshold}(\mathbf{p}, \mathbf{W}, P_{\mathsf{fp}}, n, t)$
15: $\quad\quad \mathcal{T}^\diamond \leftarrow \underset{\mathcal{T} \subset \mathcal{U}^{(t)}, |\mathcal{T}| = t}{\arg\max} s_{\mathcal{T}}$
16: $\quad\quad$ **if** $s_{\mathcal{T}^\diamond} > \tau^+$ **then**
17: $\quad\quad\quad$ **for all** $j \in \mathcal{T}^\diamond$ **and while** $\mathcal{A} = \emptyset$ **do**
18: $\quad\quad\quad\quad \mathbf{W} \leftarrow \texttt{weights}(\mathbf{y}, \mathbf{p}, \hat{\boldsymbol{\theta}}_{c_{\max}}, \mathcal{U}_{\mathsf{SI}} \cup (\mathcal{T}^\diamond \setminus \{j\}))$
19: $\quad\quad\quad\quad \tau^{+\prime} \leftarrow \texttt{threshold}(\mathbf{p}, \mathbf{W}, P_{\mathsf{fp}}, n, 1)$
20: $\quad\quad\quad\quad$ **if** $\texttt{score}(\{j\}, \Xi, \mathbf{W}) > \tau^{+\prime}$ **then** $\mathcal{A} \leftarrow \{j\}$
21: $\quad\quad\quad$ **end for**
22: $\quad\quad$ **end if**
23: $\quad$ **end while**
24: $\quad \mathcal{U}_{\mathsf{SI}} \leftarrow \mathcal{U}_{\mathsf{SI}} \cup \mathcal{A}$
25: **until** $\mathcal{A} = \emptyset$ **OR** $|\mathcal{U}_{\mathsf{SI}}| \geq c_{\max}$
26: **return** $\mathcal{U}_{\mathsf{SI}}$

The next sections describe the score computation, the pruning and the accusation, and the inference of the collusion process in more detail.

B. Score Computation

This section extends the scoring (10) of the single decoder to joint decoding. Denote by $\Xi_{\mathcal{E}}$ the set of codewords of the users of set $\mathcal{E}$. For a t-subset $\mathcal{T}$, the accusation is formulated as a hypothesis test based on the observations $(\Xi_{\mathcal{T}}, \mathbf{y}, \mathbf{p})$ and on the side-information $\mathcal{U}_{\mathsf{SI}}$ to decide between $\mathcal{H}_0$ (all $j \in \mathcal{T}$ are innocent) and $\mathcal{H}_1$ (all $j \in \mathcal{T}$ are guilty). The joint score of subset $\mathcal{T}$, $s_{\mathcal{T}} = s(\Xi_{\mathcal{T}}, \mathbf{y}, \mathbf{p}, \Xi_{\mathcal{U}_{\mathsf{SI}}})$, is just the LLR tuned on the inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ of the collusion process. This description encompasses single scores (10) for $t = 1$ (lines 6 and 20) and joint scores for $t > 1$ (line 13). In the latter case, the alternative hypothesis $\mathcal{H}_1$ should indeed be: there is at least one $j \in \mathcal{T}$ who is guilty. But this composite hypothesis test has a complexity in $O(2^t)$ per t-subset. Our approach is suboptimal for $t > 1$ but less complex.

The sequences $\mathbf{p}$, $\mathbf{y}$ and the codewords of the codebook Ξ are composed of independent random variables thanks to the code construction and the memoryless nature of the collusion. Moreover, the collusion only depends on the number of "1" symbols present in the codewords of a subset. Denote by $\boldsymbol{\varphi}$ and $\boldsymbol{\delta}$ the accumulated codewords corresponding to $\mathcal{T}$ and $\mathcal{U}_{\mathsf{SI}}$: $\boldsymbol{\varphi} =$

$\sum_{j\in\mathcal{T}} \mathbf{x}_j$ and $\boldsymbol{\delta} = \sum_{j\in\mathcal{U}_{\mathsf{SI}}} \mathbf{x}_j$. We have $\forall i \in [m], 0 \le \varphi(i) \le t$ and $0 \le \delta(i) \le n_{\mathsf{SI}}$, where $n_{\mathsf{SI}} = |\mathcal{U}_{\mathsf{SI}}|$. Thanks to the linear structure of the decoder, the score for a subset $\mathcal{T}$ of t users is simply

$$s_{\mathcal{T}} = \sum_{i=1}^{m} W(\varphi(i), i) \tag{14}$$

where $W(j,i)$ is the entry in row j and column i of a $(t+1)\times m$ weight matrix $\mathbf{W}$ which is precomputed (procedure weights() in Algorithm 1) from $(\mathbf{y},\mathbf{p})$ taking into account the side information $\mathcal{U}_{\mathsf{SI}}$ so that $\forall(\varphi,i)\in\{0,\dots,t\}\times\{1,\dots,m\}$

$$W(\varphi,i) = \log \frac{\mathbb{P}(y(i)|(\varphi,t),(\delta(i),n_{\mathsf{SI}}),p(i),\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}{\mathbb{P}(y(i)|(\delta(i),n_{\mathsf{SI}}),p(i),\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})}. \tag{15}$$

For indices s.t. $y(i)=1$, both the numerator and the denominator share a generic formula, $P(\varphi(i)+\delta(i), t+n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})$ and $P(\delta(i), n_{\mathsf{SI}}, p(i), \hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}})$, respectively, with

$$P(u,v,p,\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}) = \sum_{k=u}^{c_{\mathsf{max}}-v+u} \left(\hat{\theta}_{c_{\mathsf{max}}}(k) \cdot \binom{c_{\mathsf{max}}-v}{k-u} p^{k-u}(1-p)^{c_{\mathsf{max}}-v-k+u} \right). \tag{16}$$

In words, this expression gives the probability that $y=1$ knowing that the symbol "1" has been distributed to users with probability p, the collusion model $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, and the identity of v colluders who have u symbols "1" and $v-u$ symbols "0". For indices s.t. $y(i)=0$ in (15), the numerator and the denominator need to be "mirrored": $(P \rightarrow 1-P)$.

At iterations based on the single decoder (lines 6 and 20), $t=1$ and $\boldsymbol{\varphi}=\mathbf{x}_j$ for user j. If nobody has been deemed guilty so far, then $\delta(i) = n_{\mathsf{SI}} = 0,\ \forall i \in [m]$. This score is defined if $t + n_{\mathsf{SI}} \le c_{\mathsf{max}}$. Therefore, for a given size of side-information, we cannot conceive a score for subsets of size bigger than $c_{\mathsf{max}} - n_{\mathsf{SI}}$. This implies that in the detect-many scenario, the maximal number of iterations depends on how fast $\mathcal{U}_{\mathsf{SI}}$ grows. The procedure scores() in Algorithm 1 outputs a list of scores given a *set* of subsets of t users (which is the set of users if $t=1$ for single decoding), the code matrix and the weight matrix $\mathbf{W}$. We assume there is a deterministic way to browse all the t-subsets; in practice, this is done by the revolving door procedure [27].

C. Ranking Users and Accusation

In order to built the set $\mathcal{U}^{(t)}$, we need to rank the users based on the previous scores. We record the highest score for each user

$$s_j = \max_{\mathcal{T}: j\in\mathcal{T}} s_{\mathcal{T}}. \tag{17}$$

This step is not needed if the scores come from a single decoder. The procedure top() (line 11) ranks users according to their highest subset score and prunes the suspect list to the first $n^{(t)}$ users.

Suppose $\mathcal{T}_{\mathsf{inn}}$ is a t-subset composed of innocent users. Using rare event analysis, τ^+ is a pessimistic estimation of τ (see Section III-C) s.t. $\mathbb{P}(s(\Xi_{\mathcal{T}_{\mathsf{inn}}}, \mathbf{y}, \mathbf{p}, \Xi_{\mathcal{U}_{\mathsf{SI}}}) > \tau^+) \le \binom{n}{t}^{-1} P_{\mathsf{fp}}$. This is the procedure threshold$(\mathbf{p}, \mathbf{W}, P_{\mathsf{fp}}, n, t)$ applied with $t=1$ (lines 7 and 19) or $t \ge 1$ (line 14). Let $\mathcal{T}^\diamond$ denote the t-subset with the highest score (line 15). If $s_{\mathcal{T}^\diamond} > \tau^+$, then $\mathcal{T}^\diamond$ contains at least one colluder with a high probability. This works for any scoring function, and especially with the one explained in Section IV-B even if it is not optimal.

We accuse at most one user in $\mathcal{T}^\diamond$. We compute the following single score: $s(\mathbf{x}_j, \mathbf{y}, \mathbf{p}, \Xi_{\mathcal{U}_{\mathsf{SI}}\cup(\mathcal{T}^\diamond\setminus\{j\})})$, and we accuse user j if it is bigger than $\tau^{+\prime}$, with $\tau^{+\prime}$ s.t. $\mathbb{P}(s(\mathbf{x}_{\mathsf{inn}}, \mathbf{y}, \mathbf{p}, \Xi_{\mathcal{U}_{\mathsf{SI}}\cup(\mathcal{T}^\diamond\setminus\{j\})}) > \tau^{+\prime}) \le n^{-1} P_{\mathsf{fp}}$ (line 19). This method is suggested in [1, sec. 5.3]. The order in which we screen the users of $\mathcal{T}^\diamond$ has little importance. In practice, we focus on the users appearing more frequently in the highest subsets of (17).

D. Inference of Collusion Process

The MLE infers about the collusion process (line 4)

$$\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} = \arg\max_{\boldsymbol{\theta}\in\Theta_{c_{\mathsf{max}}}} \log \mathbb{P}(\mathbf{y}|\mathbf{p}, \mathcal{U}_{\mathsf{SI}}, \boldsymbol{\theta}). \tag{18}$$

Whenever a user is deemed guilty, the user is added to side-information and we rerun the parameter estimation to refine the collusion inference. Our *soft* decision decoding method resorts to the noise-aware collusion models (2) and (3) and sets

$$\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}} = \arg\max_{\boldsymbol{\theta}\in\{\hat{\boldsymbol{\theta}}^{(II)}_{c_{\mathsf{max}}}, \hat{\boldsymbol{\theta}}^{(I)}_{c_{\mathsf{max}}}\}} \mathbb{P}(\mathbf{y}'|\mathbf{p}, \mathcal{U}_{\mathsf{SI}}, \boldsymbol{\theta}). \tag{19}$$

This is illustrated in the right part of Fig. 3. Notice that models I and II share the same number of parameters, therefore, there is no risk of overfitting. Replacing (18) by (19) in the collusion inference step is the only change to the decoding algorithms we make to handle collusion attacks over real samples.

V. Experimental Results

We have implemented the Tardos decoders in C++.[3] Single and joint score computation is implemented efficiently using precomputed lookup tables, c.f. (14) and (15) and aggregation techniques described in [26]. For a code length of $m = 1024$ more than 10^6 single and about 10^5 joint scores, respectively, can be computed per second on a single core of a regular Intel Core2~2.6 GHz CPU. To control the runtime, the joint decoders are confined to 5-subset decoding ($t_{\mathsf{max}} = 5$) and around $p^{(t)} \approx 4.5\cdot 10^6$ computed subsets per joint decoding stage. An iterative decoding experiment can be executed on a PC within a couple of minutes, given enough memory, see [27] for details. To experimentally verify the false-positive rate controlled by rare-event analysis, up to $3\cdot 10^4$ tests per parameter setting have been performed on a cluster of PCs.

First, we compare the performance of the proposed decoders under marking assumption. Finally, we lift this unrealistic restriction and turn to a more practical assessment using soft-decision decoding.

Unless explicitly noted, the terms *single* and *joint* decoder refer to the decoders conditioned on the inference of the collusion process $\hat{\boldsymbol{\theta}}_{c_{\mathsf{max}}}$, c.f. (10) and (14). Further, we consider

[3]Source code is available at http://www.irisa.fr/texmex/people/furon/src.html.

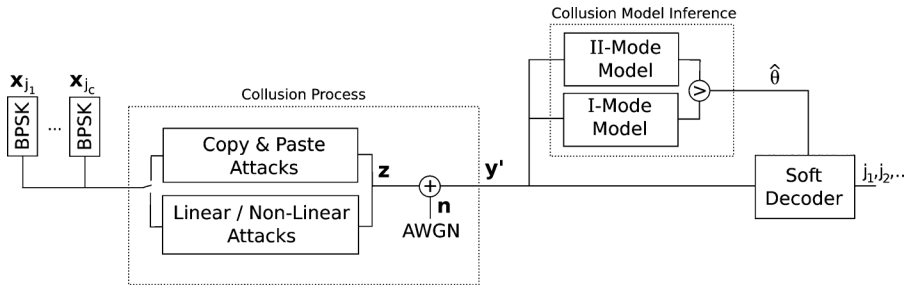

Fig. 3. Attack channel and collusion model inference.

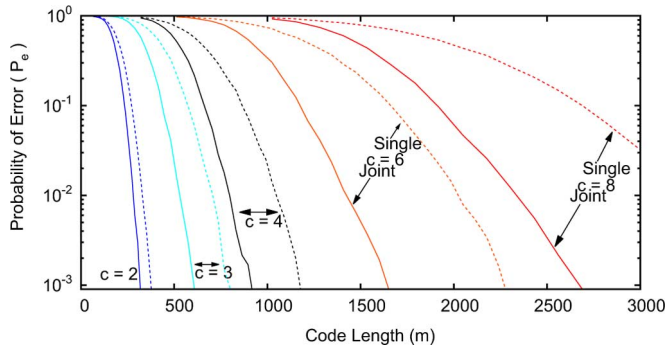

Fig. 4. Code length versus P_e for $n = 10^6$ users and different number of colluders performing *worst case* attack against a single decoder; $c_{max} = 8$.

the MAP decoders assuming knowledge of $\boldsymbol{\theta}_c$ and the compound channel decoder, c.f. (9), tuned on the worst case attack $\boldsymbol{\theta}^{\star}_{k,f_I}, \forall k \in [2,\ldots,c_{max}]$. As a baseline for a performance comparison, we always include symmetric Tardos score computation [9] with a threshold controlled by rare-event analysis (see Section III-C).

A. Decoding Performance Under Marking Assumption

1) Detect-One Scenario: Here, the aim is to catch at most one colluder—this is the tracing scenario most commonly considered in the literature. We compare our *single* and *joint* decoder performance against the results provided by Nuida *et al.* [13] (which are the best as far as we know) and, as a second reference, the symmetric Tardos decoder.

The experimental setup considers $n = 10^6$ users and $c \in \{2,3,4,6,8\}$ colluders performing *worst case* attack [22] against a single decoder. In Fig. 4, we plot the empirical probability of error $P_e = P_{fp} + P_{fn}$ obtained by running 10^4 experiments for each setting versus the code length m. The false-positive error is controlled by thresholding based on rare-event simulation, $P_{fp} = 10^{-3}$, which is confirmed experimentally. Evidently, for a given probability of error, the *joint* decoder succeeds in reducing the required code length over the *single* decoder, especially for larger collusions.

Table I compares the code length to obtain an error rate of $P_e = 10^{-3}$ for our proposed Tardos decoders and the symmetric Tardos decoder with the results reported by Nuida *et al.* [13, Table 4] under marking assumption. Except for $c = 2$, the proposed decoders can substantially reduce the required code length and the *joint* decoder improves the results of the *single* decoder. Note that Nuida's results give analytic code length assuming a particular number of colluders for constructing the code but the collusion attack is arbitrary (i.e., not necessary fair) whereas our results are experimental estimates based on worst case attack against a single decoder and without knowing c at the code construction. Results with c known are provided in [27] and show a slightly better performance: the required code length of the *joint* decoder is then slightly shorter than Nuida's code in case $c = 2$.

TABLE I
CODE LENGTH COMPARISON FOR DETECT-ONE SCENARIO: $n = 10^6$, WORST CASE ATTACK AGAINST A SINGLE DECODER, $P_e = 10^{-3}$

Colluders (c)	Nuida *et al.* [13]	Symm. Tardos	Proposed ($c_{max} = 8$) Single	Joint
2	253	~ 416	~ 368	~ 304
3	877	~ 864	~ 776	~ 584
4	1454	~ 1472	~ 1152	~ 904
6	3640	~ 2944	~ 2304	~ 1616
8	6815	~ 5248	~ 3712	~ 2688

2) Detect-Many Scenario: We now consider the more realistic case where the code length m is fixed and the false-negative error rate is only a minor concern[4] while the false-positive probability is critical to avoid an accusation of an innocent. The aim is to identify as many colluders as possible.

Fig. 5(a) and (b) shows the average number of identified colluders by different decoding approaches. The experimental setup considers $n = 10^6$ users, code length $m = 2048$, and several collusion attacks (*worst case*, i.e., minimizing the achievable rate of a single or joint decoder, *interleaving* and *majority* which is a rather mild attack) carried out by two to eight colluders. The global probability of a false positive error is fixed to $P_{fp} = 10^{-3}$.

As expected, the MAP single decoder knowing $\boldsymbol{\theta}_c$ provides the best decoding performance among the single decoders, yet is unobtainable in practice. The symmetric Tardos decoder performs poorly but evenly against all attacks; the single decoder based on the compound channel (9) improves the results only slightly.

The *joint* decoders consistently achieve to identify most colluders—with a dramatic margin in case the traitors choose the worst case attack against a single decoder. This attack bothers the very first step of our decoder, but as soon as some side information is available or a joint decoder is used, this is no longer the worst case attack. Finding the worst case attack against our iterative decoder is indeed difficult. A good guess is the interleaving attack which is asymptotically the worst case against the joint decoder [2]. The experiments show that it reduces the performance of the *joint* decoders substantially for large c.

[4]A tracing scheme rightly accusing a colluder half of the time might be enough to dissuade dishonest users.

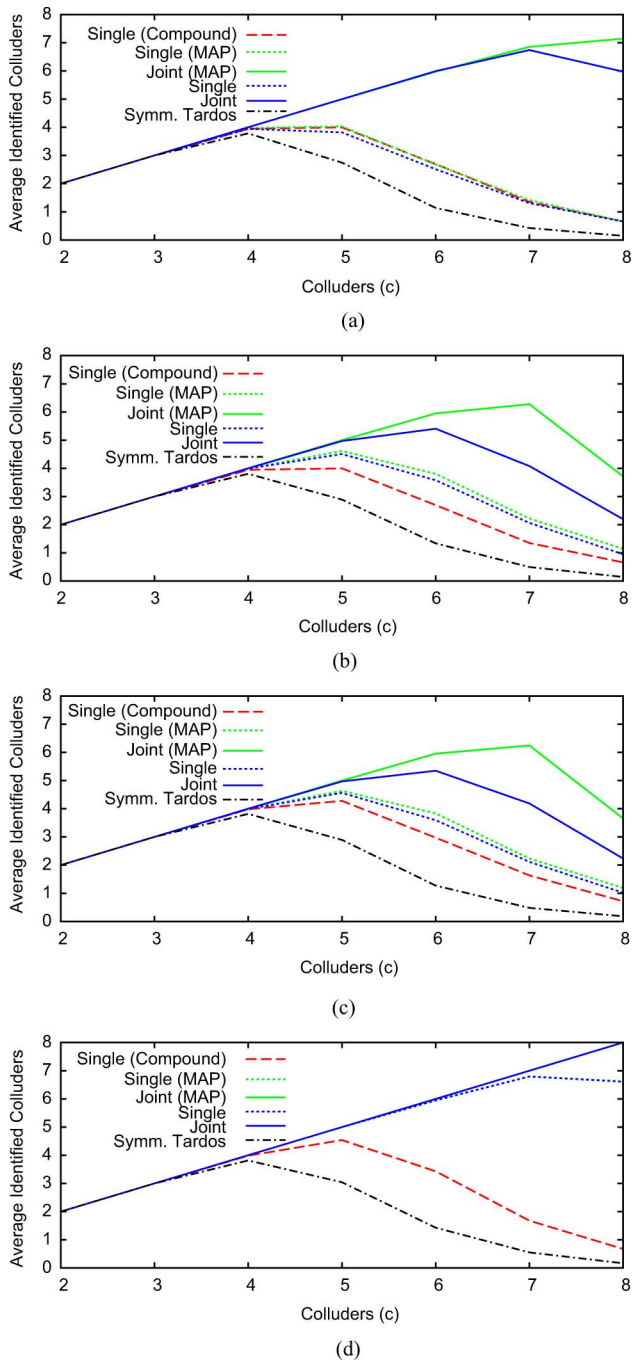

Fig. 5. Decoder comparison in detect-many tracing scenario: $n = 10^6$, $m = 2048$, $P_{\text{fp}} = 10^{-3}$, $c_{\max} = 8$. (Best viewed in color.) (a) *Worst case* attack against single decoder. (b) *Worst case* attack against joint decoder. (c) *Interleaving* attack. (d) *Majority* attack.

The decoder based on the inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ and the true MAP are different when c is lower than $c_{\max}$. However, this is not a great concern in practice for a fixed m: for small c, the code is long enough to face the collusion even if the score is less discriminative than the ideal MAP; for big c the score of our decoder gets closer to the ideal MAP.

B. Decoding Performance of Soft Decoder

We assess the performance of the soft decision decoders proposed in Section II-C in two tracing scenarios: i) Kuribayashi considers in [14] $n = 10^4$ users and code length $m = 10^4$; and ii) a large-scale setup with 33 554 432 users and $m = 7440$ where Jourdas and Moulin [16] provide results for their high-rate random-like fingerprinting code under averaging and interleaving attack.

In Fig. 6, we compare the average number of identified colluders for the *single* and *joint* decoder using different estimates of the collusion process. A simple approach, termed *hard* decision decoding in the sequel, first thresholds $\mathbf{y}'$ (to quantize $y'(i)$ into 0 if $y'(i) < 0$ and 1 otherwise), and then employs the collusion process inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ of (18) on the hard outputs. The term *soft* relates to the noise-aware decoders relying on $\hat{\boldsymbol{\theta}}^{(I)}_{c_{\max}}$ or $\hat{\boldsymbol{\theta}}^{(II)}_{c_{\max}}$ chosen adaptively based on the likelihood of the two models [see (19)]. All plots also show the results for the (hard-thresholding) symmetric Tardos decoder. The false-positive rate is set to 10^{-4}. Extensive experiments ($3 \cdot 10^4$ test runs) have been carried out to validate the accusation threshold obtained by rare-event simulation. As expected, soft decoding offers substantial gains in decoding performance. The margin between the *single* and *joint* decoders depends on the collusion strategy. Dramatic improvements can be seen when the collusion chooses the *worst case* attack against a single decoder, c.f. Fig. 6(a). On the other hand, the gain is negligible when averaging is performed.

Note that the attacks in (a)–(c) pertain to the copy-and-paste attacks while Fig. 6(d) shows the linear *averaging* attack.

Comparison with the results provided in [14] for the *majority* attack is difficult: 1) they were obtained for Nuida's discrete code construction [13] tuned on $c = 7$ colluders and 2) the false-positive rate of [14] does not seem to be under control for the symmetric Tardos code. We suggest using the *hard* symmetric Tardos decoder [9] as a baseline for performance comparison. By replacing the accusation thresholds proposed in [14] with a rare-event simulation, we are able to fix the false-alarm rate in case of the symmetric Tardos code. Furthermore, the decoding results given in [14] for the discrete variant of the fingerprinting code (i.e., Nuida's construction) could be significantly improved by rare-event simulation-based thresholding. Contrary to the claim of [14], *soft* decision decoding always provides a performance benefit over the *hard* decoders.

In Fig. 7 we illustrate the decoding performance when dealing with a large user base. We consider *averaging* and *interleaving* attacks by $c = 2, \ldots, 12$ and $c = 2, \ldots, 8$ colluders ($c_{\max} = 12$ and $c_{\max} = 8$, respectively) followed by AWGN with variance $\sigma_{\text{n}}^2 = 1$. The global false-positive rate is set to 10^{-3}. The benefit of the *soft* decoding approach is clearly evident. Joint decoding provides only a very limited increase in the number of identified colluders. For comparison, Jourdas and Moulin indicate an error rate of $P_{\text{e}} = 0.0074$ for $c = 10$ colluders in the first, and $P_{\text{e}} = 0.004$ for $c = 5$ colluders in the second setting for a detect-one scenario [16].

In [28], $P_{\text{fp}} = 0.0016$ and $P_{\text{fn}} = 0.044$ are given for the first experiment [Fig. 7(a)] by introducing a threshold to control the false-positive rate. Our *soft joint* decoder achieves a $P_{\text{fn}} = 0.046$ for $P_{\text{fp}} = 10^{-3}$ (for $c = 10$ colluders), catching 2.6 traitors on average.

In the second experiment [see Fig. 7(b)], our *joint* decoder compares more favorably: with the given code length, all $c = 5$ colluders can be identified and for a collusion size $c = 8$,

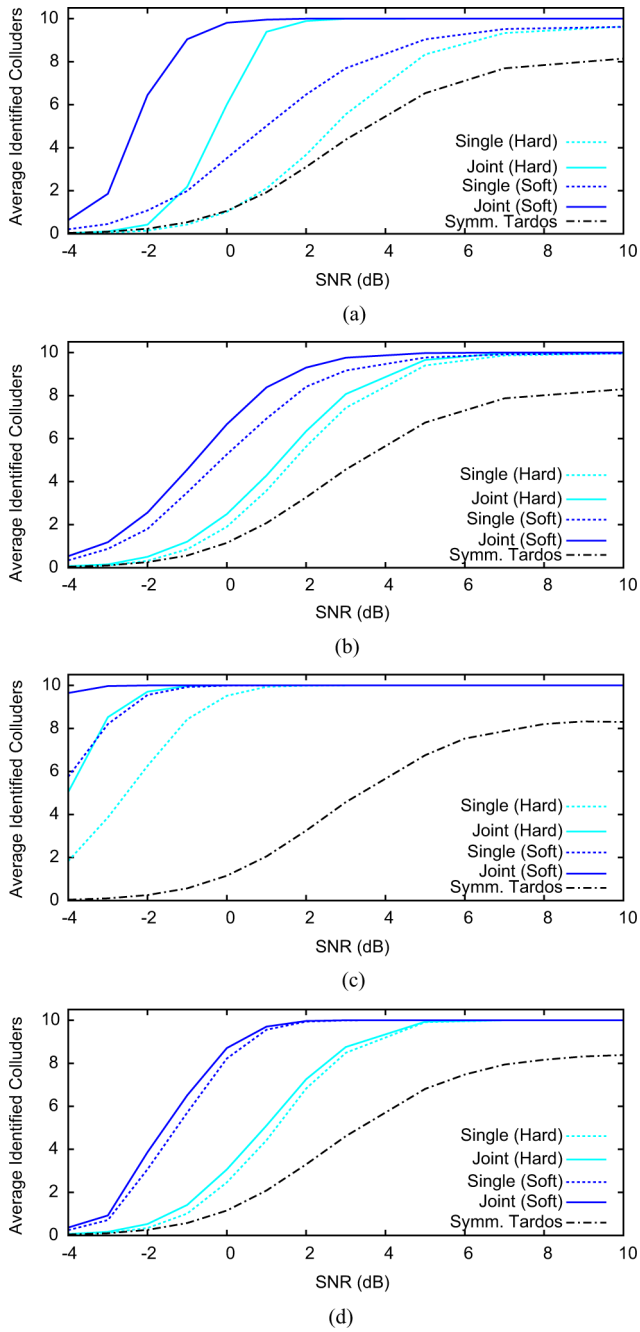

Fig. 6. Kuribayashi setup: $n = 10^4$, $m = 10^4$, $P_{\text{fp}} = 10^{-4}$, $c = 10$, $c_{\max} = 20$; *worst case*, *interleaving*, *majority* and *averaging* attack followed by AWGN ($-4, \ldots, 10$ dB SNR). (a) *Worst case* attack against single decoder. (b) *Interleaving* attack. (c) *Majority* attack. (d) *Averaging* attack.

4.5 traitors are accused without observing any decoding failure in $3 \cdot 10^3$ tests.

C. Runtime Analysis

Single decoding can be efficiently implemented to compute more than one million scores for a code of length $m = 1024$ per second. Its complexity is in $O(m \cdot n)$. Selecting the $n^{(t)}$ most likely guilty users can be efficiently done with the max-heap algorithm. Yet, it consumes a substantial part of the runtime for small m. The runtime contribution of score computation for the joint decoding stages is in $O(m \cdot p^{(t)})$ and clearly depends on

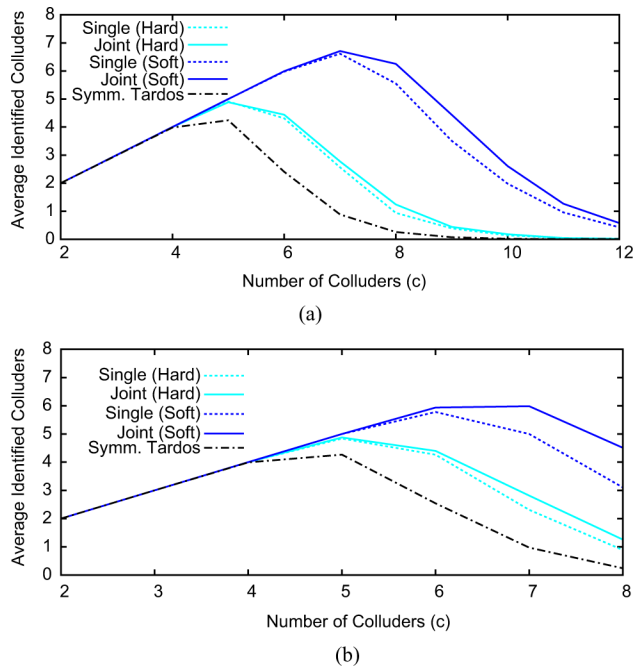

Fig. 7. Jourdas and Moulin setup: $n = 33\,554\,432$, $m = 7\,440$, $P_{\text{fp}} = 10^{-3}$, *averaging* and *interleaving* attack followed by AWGN (0 dB SNR. (a) *Averaging* attack. (b) *Interleaving* attack.

the size of the pruned list of suspects. However, the computation of the score per subset is independent of the subset size t thanks to the *revolving door* enumeration method of the subsets.[5] Restricting $p^{(t)}$ and $t_{\max}$ keeps the joint decoding approach computationally tractable. Better decoding performance can be obtained using higher values at the cost of a substantial increase in runtime. Experiments have shown that even the moderate settings ($p^{(t)} \approx 4.5 \cdot 10^6$ and $t_{\max} = 5$) achieve a considerable gain of the joint over the single decoder for several collusion channels.

Thresholding accounts for more than half of the runtime in the experimental setups investigated in this work. However, this is not a serious issue for applications with a large user base or when $p^{(t)}$ becomes large. Thresholding depends on the subset size t because a large number of random codeword combinations must be generated and because we seek lower probability level in $O(P_{\text{fp}}/n^t)$. Therefore, the complexity is in $O(m \cdot t^2 \cdot \log n)$ according to [25]. There are no more than $c_{\max}$ such iterations with $t \leq c_{\max}$, so that the complexity of the thresholding is in $O(m \cdot \log n)$ and the global complexity of our decoder stays in $O(m \cdot n)$.

More details about the runtime of our implementation are given in [27]. Note that results have been obtained with a single CPU core although a parallel implementation can be easily achieved.

VI. Conclusion

Decoding fingerprinting codes in practice means to trace guilty persons over a large set of users while having no information about the size nor the strategy of the collusion. This

[5]In each step φ is updated by replacing one user's codeword. See [27] for details.

must be done reliably by guaranteeing a controlled probability of false alarm.

Our decoder implements provably good concepts of information theory (joint decoding, side information, linear decoder for compound channels) and statistics (estimation of extreme quantile of a rare event). Its extension to soft output decoding is straightforward as it does not change the architecture. Very competitive results have been obtained experimentally.

Since the proposed iterative method is neither just a single decoder nor completely a joint decoder (it only considers subsets over a short list of suspects), it is rather difficult to find the best distribution for code construction and its worst case attack. Experiments show that the interleaving attack is indeed more dangerous than the worst case attack against a single decoder.

APPENDIX

We prove that $\mathcal{E}_{c_{\max}}(\boldsymbol{\theta}_c) = \{\tilde{\boldsymbol{\theta}}_k | k \leq c_{\max}, \mathbb{P}(y|p,\tilde{\boldsymbol{\theta}}_k) = \mathbb{P}(y|p,\boldsymbol{\theta}_c),\ \forall (y,p) \in \{0,1\}\times[0,1]\}$ is one sided w.r.t. some $p \in [0,1]$ and thus w.r.t. the expectation over f if $f(p) > 0$ for at least one of these values of p. The collusion channels of this set share the property that $\mathbb{P}(Y=1|p,\tilde{\boldsymbol{\theta}}_k) = q(p) \geq 0, \forall p \in [0,1]$. From [22, eq. (20)]

$$\mathbb{P}(Y=1|X=1,p,\tilde{\boldsymbol{\theta}}_k) = q(p) + k^{-1}(1-p)q'(p) \quad (20)$$

$$\mathbb{P}(Y=1|X=0,p,\tilde{\boldsymbol{\theta}}_k) = q(p) - k^{-1}pq'(p). \quad (21)$$

Take $(\tilde{\boldsymbol{\theta}}_{k_A}, \tilde{\boldsymbol{\theta}}_{k_B}) \in \mathcal{E}_{c_{\max}}(\boldsymbol{\theta}_c)^2$ s.t. $k_A < k_B$. We first show that $R(f,\tilde{\boldsymbol{\theta}}_{k_A}) \geq R(f,\tilde{\boldsymbol{\theta}}_{k_B})$ so that $\tilde{\boldsymbol{\theta}}_{c_{\max}}$ is a minimizer of $R(f,\boldsymbol{\theta})$ over $\mathcal{E}_{c_{\max}}(\boldsymbol{\theta}_c)$. Denote by (μ_1,μ_2) the following conditional probability distributions:

$$\mu_1(y,x|p) = \mathbb{P}(Y=y|p) = q(p)^y(1-q(p))^{(1-y)} \quad (22)$$

$$\mu_2(y,x|p) = \mathbb{P}(Y=y|X=x,p,\tilde{\boldsymbol{\theta}}_{k_A}). \quad (23)$$

Then, $\mathbb{P}(Y|X,p,\tilde{\boldsymbol{\theta}}_{k_B}) = (1-\lambda)\mu_1(Y,X|p) + \lambda\mu_2(Y,X|p)$, $\forall p \in [0,1]$, with $\lambda = k_A/k_B < 1$. The mutual information is a convex function of $\mathbb{P}(Y|X,p)$ for fixed $\mathbb{P}(X|p)$ so that, once integrated over $f(p)$, we have

$$R(f,\tilde{\boldsymbol{\theta}}_{k_B}) \leq (1-\lambda)\cdot 0 + \lambda \cdot R(f,\tilde{\boldsymbol{\theta}}_{k_A}) \leq R(f,\tilde{\boldsymbol{\theta}}_{k_A}). \quad (24)$$

The second inequality turns to be an equality if only if $R(f,\tilde{\boldsymbol{\theta}}_{k_A}) = 0$. It means that k_A colluders succeed to nullify the mutual information between X and Y for any p s.t. $f(p) > 0$. Then, $\tilde{\boldsymbol{\theta}}_{c_{\max}}$ is not a unique minimizer. This can happen if k_A is big enough, but it is impossible for distributions s.t. $f(p) > 0$ for some $p < 1/k_A$, see [22, sec. 4]. This especially holds for f_T.

We now prove that (8) holds $\forall \boldsymbol{\theta} \in \mathcal{E}_{c_{\max}}(\boldsymbol{\theta}_c)$. This is equivalent to

$$R(f,\tilde{\boldsymbol{\theta}}_k) - D(\mathbb{P}(Y,X|\tilde{\boldsymbol{\theta}}_k)||\mathbb{P}(Y,X|\tilde{\boldsymbol{\theta}}_{c_{\max}})) - R(f,\tilde{\boldsymbol{\theta}}_{c_{\max}}) \geq 0 \quad (25)$$

where the LHS is of the form $\mathbb{E}_{P\sim f}[g(P)]$ with $g(0) = g(1) = 0$. After developing the expressions for $0 < p < 1$, we find that

$$\begin{aligned} g(p) = (k^{-1} - c_{\max}^{-1})p(1-p)\cdot & \\ \Bigg(q'(p)\log\left(1 + \frac{1-p}{c_{\max}}\frac{q'(p)}{q(p)}\right) & \\ + q'(p)\log\left(1 + \frac{p}{c_{\max}}\frac{q'(p)}{1-q(p)}\right) & \\ - q'(p)\log\left(1 - \frac{1-p}{c_{\max}}\frac{q'(p)}{1-q(p)}\right) & \\ - q'(p)\log\left(1 - \frac{p}{c_{\max}}\frac{q'(p)}{q(p)}\right)\Bigg). & \end{aligned} \quad (26)$$

The four terms inside parenthesis are not negative because $x\log(1+\gamma x) \geq 0$ for $\gamma > 0$ and $x > -\gamma^{-1}$. Since $k < c_{\max}$, we obtain $g(p) \geq 0$ and (8) after expectation over f.

ACKNOWLEDGMENT

The authors are grateful for the efforts of the anonymous reviewers who contributed by their valuable comments to the quality of this paper.

REFERENCES

[1] P. Moulin, Universal Fingerprinting: Capacity and Random-Coding Exponents arXiv:0801.3837v3, 2011 [Online]. Available: http://arxiv.org/abs/0801.3837

[2] Y.-W. Huang and P. Moulin, "On the saddle-point solution and the large-coalition asymptotics of fingerprinting games," *IEEE Trans. Inform. Forensics Security*, vol. 7, no. 1, pp. 160–175, Feb. 2012.

[3] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms, SODA '09*, New York, Jan. 2009, pp. 336–345.

[4] N. P. Anthapadmanabhan, A. Barg, and I. Dumer, "On the fingerprinting capacity under the marking assumption," *IEEE Trans. Inform. Theory*, vol. 54, no. 6, pp. 2678–2689, Jun. 2008.

[5] E. Abbe and L. Zheng, "Linear universal decoding for compound channels," *IEEE Trans. Inform. Theory*, vol. 56, no. 12, pp. 5999–6013, Dec. 2010.

[6] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 1–24, May 2008.

[7] E. Amiri, "Fingerprinting codes: Higher rates, quick accusations," Ph.D. dissertation, Simon Fraser Univ., Burnaby, BC, Canada, 2010.

[8] K. Nuida, "Short collusion-secure fingerprint codes against three pirates," in *Proc. Information Hiding Workshop, IH '10, ser. Lecture Notes in Computer Science*, Calgary, Canada, Oct. 2010, vol. 6387, pp. 86–102.

[9] B. Skoric, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes Cryptography*, vol. 46, no. 2, pp. 137–166, Feb. 2008.

[10] T. Furon and L. Pérez-Freire, "EM decoding of Tardos traitor tracing codes," in *Proc. ACM Multimedia Security Workshop*, Princeton, NJ, Sep. 2009, pp. 99–106.

[11] M. Fernandez and M. Soriano, "Identification of traitors in algebraic-geometric traceability codes," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 3073–3077, Oct. 2004.

[12] A. Barg, G. R. Blakley, and G. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Trans. Inform. Theory*, vol. 49, no. 4, pp. 852–865, Apr. 2003.

[13] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of discrete Tardos fingerprinting codes," *Designs, Codes Cryptography*, vol. 52, no. 3, pp. 339–362, Mar. 2009.

[14] M. Kuribayashi, "Experimental assessment of probabilistic fingerprinting codes over AWGN channel," in *Proc. 5th Int. Workshop Security, IWSEC '10, ser. Lecture Notes in Computer Science*, Kobe, Japan, Nov. 2010, vol. 6432, pp. 117–132.
[15] H. G. Schaathun, "On error-correcting fingerprinting codes for use with watermarking," *Multimedia Systems*, vol. 13, no. 5, pp. 331–344, 2008.
[16] J.-F. Jourdas and P. Moulin, "High-rate random-like spherical fingerprinting codes with linear decoding complexity," *IEEE Trans. Inform. Forensics Security*, vol. 4, no. 4, pp. 768–780, Dec. 2009.
[17] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1897–1905, Sep. 1998.
[18] B. Skoric, S. Katzenbeisser, H. Schaathun, and M. Celik, "Tardos fingerprinting codes in the combined digit model," *IEEE Trans. Inform. Forensics Security*, vol. 6, no. 3, pp. 906–919, 2011.
[19] L. Pérez-Freire and T. Furon, "Blind decoder for binary probabilistic traitor tracing codes," in *Proc. First IEEE Int. Workshop Information Forensics Security. WIFS'09*, London, U.K., Dec. 2009, pp. 56–60.
[20] T. Furon, A. Guyader, and F. Cérou, "On the design and optimisation of Tardos probabilistic fingerprinting codes," in *Proc. 10th Information Hiding Workshop, ser. Lecture Notes in Computer Science*, Santa Barbara, CA, May 2008, pp. 341–356.
[21] A. Somekh-Baruch and N. Merhav, "On the capacity game of private fingerprinting systems under collusion attacks," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 884–899, May 2005.
[22] T. Furon and L. Pérez-Freire, "Worst case attacks against binary probabilistic traitor tracing codes," in *Proc. First IEEE Int. Workshop Information Forensics and Security WIFS'09*, London, U.K., Dec. 2009, pp. 46–50.
[23] Y.-W. Huang and P. Moulin, "Capacity-achieving fingerprint decoding," in *Proc. IEEE Int. Workshop Information Forensics and Security, WIFS '09*, London, U.K., Dec. 2009, pp. 51–55.
[24] A. Simone and B. Skoric, "Accusation probabilities in Tardos codes: Beyond the Gaussian approximation," *Designs, Codes Cryptography*, vol. 63, no. 3, pp. 379–412, 2012.
[25] A. Guyader, N. Hengartner, and E. Matzner-Lober, "Simulation and estimation of extreme quantiles and extreme probabilities," *Applied Math. Optimization*, vol. 64, no. 2, pp. 171–196, 2011.
[26] P. Meerwald and T. Furon, "Iterative single Tardos decoder with controlled probability of false positive," in *Proc. IEEE Int. Conf. Multimedia Expo, ICME '11*, Barcelona, Spain, Jul. 2011, pp. 1–6.
[27] P. Meerwald and T. Furon, "Towards joint Tardos decoding: The 'Don Quixote' algorithm," in *Proc. Information Hiding Conf., IH ' 11, ser. Lecture Notes in Computer Science*, Prague, Czech Republic, May 2011, vol. 6958, pp. 28–42.
[28] J.-F. Jourdas and P. Moulin, "A high-rate fingerprinting code," in *Proc. IS&T/SPIE Symp. Electronic Imaging, Security, Forensics, Steganography Watermarking of Multimedia Contents X*, San Jose, CA, Jan. 2008.

Peter Meerwald received the M.S. degree in computer sciences from Bowling Green State University, OH, in 1999, and the M.S. and Ph.D. degrees from the University of Salzburg, Austria, in 2001 and 2010.

From 2001 to 2007, he was a Research Engineer with Sony DADC, Salzburg, Austria, working on software security and copy protection. In 2010 and 2011, he pursued postdoctoral research on multimedia fingerprinting at Inria Research Center, Rennes, France. At present, he is a Research Engineer with BCT Electronic, Salzburg, Austria, developing audio processing and communication systems on embedded Linux platforms.

Teddy Furon received the M.S. degree in digital communications and the Ph.D. degree in signal and image processing from the Ecole Nationale Supérieure des Télécommunications de Paris, Paris, France, in 1998 and 2002, respectively.

From 1998 to 2001, he was a Research Engineer with the Security Lab of Thomson, Rennes, France, working on digital watermarking in the framework of copy protection. He continued working on digital watermarking as a Postdoctoral Researcher at the TELE Lab of the Université Catholique de Louvain, Louvain, Belgium. He also worked in the Security Lab of Technicolor. He is at present a Researcher working within the TEXMEX Team-Project in the Inria Research Center, Rennes, France.

Dr. Furon serves as Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, of the EURASIP *Journal on Information Security*, of the IET *Journal of Information Security*, and of the *Elsevier Digital Signal Processing Journal*.