Impacts of Watermarking Security on Tardos-based Fingerprinting

Benjamin Mathon, Patrick Bas, François Cayre, Benoît Macq

Impacts of Watermarking Security on Tardos-based Fingerprinting

Benjamin Mathon*[1], Patrick Bas†[2], François Cayre‡[1], and Benoît Macq§[3]

[1]GIPSA-Lab, Institut Polytechnique de Grenoble
[2]LAGIS, École Centrale de Lille
[3]TELE, Université catholique de Louvain

Abstract

This article presents a study of the embedding of Tardos binary fingerprinting codes with watermarking techniques. By taking into account the security of the embedding scheme, we present a new approach for colluding strategies which relies on the possible estimation error rate of the code symbols (denoted ϵ). We derive a new attack strategy called "ϵ-Worst Case Attack" and show its efficiency using the computation of achievable rates for simple decoding. Then we consider the interplay between security and robustness regarding the accusation performances of the fingerprinting scheme and show that (1) for a same accusation rate secure schemes can afford to be less robust than insecure ones, and (2) that secure schemes enable to cast the Worst Case Attack into an interleaving attack. Additionally, we use the security analysis of the watermarking scheme to derive from ϵ a security attack for a fingerprinting scheme based on Tardos codes and a new scheme called stochastic spread-spectrum watermarking. We compare a removal attack against an AWGN robustness attack and we show that for the same distortion, the combination of a fingerprinting attack and a security attack easily outperform classical attacks even with a small number of observations.

Keywords: Active Fingerprinting; Security; Spread-spectrum; Watermarking

*benjamin.mathon@grenoble-inp.fr
†patrick.bas@ec-lille.fr
‡francois.cayre@grenoble-inp.fr
§benoit.macq@uclouvain.be

1 Introduction

Active fingerprinting (also known as traitor tracing) addresses the piracy of intellectual property on digital contents. This field of study consists in marking each numerical copy of a multimedia content with the unique identifier (a fingerprint sequence) of the customer. If the copy is found on illegal networks, a distributor can afterward trace the user responsible of the leak. Active fingerprinting has been first generalized by Wagner [1] where fingerprints are defined as *characteristics of an object that tend to distinguish it from other similar objects*. The author gives historical examples of fingerprints used for tracing illegal copies of different objects such as human fingerprints or copyrights on logarithm tables (generated by modifying the least significant digits [2]).

If a malicious user (an adversary) can exactly extract all the digits of his sequence, he will then be able to modify or erase the fingerprint and not be accused if the content is found on illegal networks. The localization of the tracing digits in the contents is secret and has to be only known by the distributor. However, several adversaries can work together (they form a collusion and each adversary is called a colluder) by pooling their own version of the multimedia content, and can attempt to estimate the fingerprint positions. In fact, where they see a difference of digits in the content, they can first infer the positions used for tracing. Then, they can copy the digit of one member of the collusion on each location in order to forge a pirated copy. Fingerprinting methods which resist collusion attacks have been first studied by Blackley *et. al.* [3].

Most fingerprinting techniques efficiently accuse at least one member of the collusion under the *marking assumption* [4]: the collusion only modifies the digits at positions where they see differences during the construction of the pirated sequence. In this article, we are interested in binary Tardos probabilistic fingerprint codes [5]. The sequences of length m are generated according to a stochastic process for n users, taking into account a false alarm probability p_{fa} (probability of accusing at least one innocent user). These codes offer an optimal solution for collusion-secure fingerprinting under the marking assumption because the length m meets the Peikert's theoretical lower bound [6]: $m \sim O(c^2 \log(n/p_{fa}))$ where c is the maximal size of the collusion. Because of their optimal asymptotic length, Tardos codes had a important impact in the community and several works concerning the reduction of the length of these codes were published [7–10].

Recent works [11, 12] propose a collusion attack for Tardos codes called Worst Case Attack (WCA) which minimizes the mutual information between the sequence of one colluder and the pirated sequence forged by the collusion.

Watermarking techniques embed the codes into context because they are particularly robust against manipulations of contents such as noise addition, compression or geometrical modifications. The distributor reads wa-

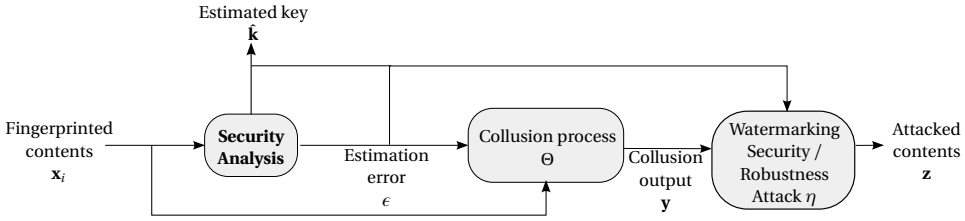


Figure 1: Security issues in Fingerprinting: the adversary can perform attacks on the fingerprinting process and the watermarking channel by dedicated attacks or robustness attacks. ϵ denotes the estimation error rate of each symbol and η the bit error rate generated by the robustness or security attack.

termarked fingerprinting codes and afterwards runs the accusation process.

While several practical implementations with these robustness aspects have been studied [13–15], we consider in this article the constraint of watermarking security which refers to *"the inability by unauthorized users to have access to the raw watermarking channel"* [16]. Following a cryptographic model, security in watermarking is generally based on Kerckhoffs' principle [17] and relates to the use of a secret for the embedding and the decoding of the sequences. According to the quality of estimation of the secret key, adversaries read, modify or erase the hidden sequences. The estimation of the secret key can be done using marked contents with involved techniques like Principal Component Analysis (PCA) [18], Independent Component Analysis (ICA) [19] or clustering [20]. There are two principal differences between robustness and security. On one hand, robustness attacks are not intentional: signal processing operations on watermarked contents like compression can be done by the provider before the legal use by a user of this content. On the contrary, security attacks come from an adversary who wants to hack the watermarking scheme, such attacks are more aggressive because if the scheme is not secure, the adversary can both alter the embedded message and perform an optimal minimization of the attack distortion [19].

In this article, we propose a new fingerprinting collusion attack called ϵ-WCA which takes into account the security of the watermarking scheme via the estimation error rate ϵ. We show the interplay between fingerprinting collusions and watermarking security attacks and we compare the impacts of a security attack and of a robustness attack for spread-spectrum watermarking. Figure 1 illustrates the global framework of our study: the embedding process undergoes a security analysis in order to estimate both the embedding symbols with an estimation error rate ϵ and the estimated secret embedding key $\hat{\mathbf{k}}$. ϵ is afterwards taken into account by the set of colluders in order to perform a collusion attack. Finally, the collusion perform a security attack (using the estimation of the key) or a robustness attack to alter the

embedded fingerprinting code.

This work shows that secure embedding schemes enables to boil down the Worst Case Attack into an interleaving attack. Additionally, for a given accusation rate, secure schemes can undergo a lower SNR than insecure ones. The last important contribution of the article is that watermark security attacks can outperform AWGN attacks even when the estimation error rate is big. This work is an extension of the work presented in [21], where the ϵ-WCA was introduced, and takes into account the interplay between fingerprinting constraints (accusation performance) and watermarking constraints (security and robustness).

This article is organized as follows. Section 2 presents the mathematical notations used throughout the article. Section 3 recalls Tardos fingerprinting codes and the colluding strategy. Next, in the section 4, we present the "ϵ-Worst Case Attack" taking into account the security of the watermarking scheme (probability of ϵ for the adversaries to decode a wrong symbol). We evaluate this strategy in term of accusation for a simple fingerprinting decoder. The robustness constraint for watermarking is modeled by a Binary Symmetric Channel (BSC) with probability η and we study the impact of security and robustness on the accusation process. Finally, in section 5, we present a practical scheme for fingerprinting using spread-spectrum watermarking. An adversary practically estimates the secret key and, after computing the ϵ-WCA with others members of the collusion, attempts to modify the pirated sequence by subtracting the estimated watermark signal. The performances of this security attack vs. the AWGN channel are finally compared.

2 Notations

We first list the notations and conventions used in this article. Functions are denoted in roman fonts, sets in calligraphy fonts, vectors and matrices in bold fonts and variables in italic fonts. Vectors are written in small letters and matrices in capital ones. The content of a vector $\mathbf{x}$ with length n is denoted by $(\mathbf{x}(0)\dots\mathbf{x}(n-1))$. $H(X)$ and $H_b(p)$ denote respectively the entropy of X and the entropy of $X \sim \mathcal{B}(p)$. $P_X(.)$ denotes the p.d.f. of a random variable X. $\langle . \rangle$ denotes the usual scalar product and $\|.\|_2$ is the Euclidean norm. $k \bmod n$ is k modulo n. $\mathcal{M}_{n,m}(\mathbb{R})$ is the set of matrices of size $m \times n$ over the field $\mathbb{R}$. $[n]$ and $[\![0, n-1]\!]$ both denote the set of integers $\{0, 1, \dots, n-1\}$.

3 Tardos' fingerprint codes and attack strategies

3.1 Reminders on Tardos fingerprint codes

Tardos codes [5] are very popular thanks to their minimal asymptotic length. In this subsection, we recall the construction of the code for n users resistant against collusions of c adversaries.

A fingerprinting code is represented by a matrix $\mathbf{B} \in \mathcal{M}_{n,m}(\mathbb{F}_2)$. Each row $\mathbf{b}_j$ of $\mathbf{B}$ is a fingerprint sequence of m bits which will be used in order to identify the user $j \in [n]$. Columns of $\mathbf{B}$ (i-th symbols of the sequences) are generated according to a Bernoulli distribution:

$$\forall j \in [n],\ \forall i \in [m],\ \mathbf{B}(j,i) \sim \mathcal{B}(p_i). \tag{1}$$

Variables p_i are distributed in the set $[t, 1-t]$ ($t = 1/(300c)$), according to a random variable P with p.d.f.:

$$\mathrm{f}_P(p) = \left((\pi - 4t')\sqrt{p(1-p)}\right)^{-1}, \tag{2}$$

with $t' = \arcsin\sqrt{t}$. In the collusion attack framework, a collusion $\mathcal{C} = \{j_0, \ldots, j_{c-1}\} \subset [n]$ of c adversaries creates a pirated fingerprint sequence $\mathbf{b}$ of m bits by mixing the symbols of their respective sequences according to a specific strategy. Formally, the pirated sequence is:

$$\mathbf{b} = \left(\mathbf{b}_{k_0}(0) \ldots \mathbf{b}_{k_{m-1}}(m-1)\right), \tag{3}$$

with:

$$(k_0 \ldots k_{m-1}) \in \mathcal{C}^m. \tag{4}$$

Tardos accusation process works under the *marking assumption*. It means that each symbol $\mathbf{b}(i)$ of the pirated sequence is chosen among the bits of the colluders. With this assumption, if they all agree with the symbol "1" (resp. "0") for a position $i \in [m]$, the symbol $\mathbf{b}(i)$ will be a "1" (resp. "0"). Eq. (3) respects this condition. Note that we do not consider unreadable digits here, this assumption is motivated by the spread-spectrum technique in Section 5 where decoded symbols are always "0" or "1".

The goal of the distributor of multimedia contents is to accuse at least one member of the collusion given a false alarm probability p_{fa} (the probability that at least one innocent is accused) and a error probability (probability that a colluder is not accused). Tardos accusation process (improved by Škorić *et. al.* [22]) implies the construction of a matrix $\mathbf{U} \in \mathcal{M}_{n,m}(\mathbb{R})$:

$$\mathbf{U}(j,i) = \begin{cases} \mathrm{g}_1(p_i), & \text{if } \mathbf{b}(i)=1, \mathbf{b}_j(i)=1, \\ \mathrm{g}_0(p_i), & \text{if } \mathbf{b}(i)=1, \mathbf{b}_j(i)=0, \\ \mathrm{g}_0(1-p_i), & \text{if } \mathbf{b}(i)=0, \mathbf{b}_j(i)=1, \\ \mathrm{g}_1(1-p_i), & \text{if } \mathbf{b}(i)=0, \mathbf{b}_j(i)=0, \end{cases} \tag{5}$$

given:

$$\mathrm{g}_1(p) = \sqrt{\frac{1-p}{p}},\ \mathrm{g}_0(p) = -\sqrt{\frac{p}{1-p}}. \tag{6}$$

The accusation score of a user $j \in [n]$ is given by:

$$\mathrm{S}_j = \sum_{i=0}^{m-1} \mathbf{U}(j,i). \tag{7}$$

A user j is accused of participating in the creation of a pirated sequence $\mathbf{b}$ if $\mathrm{S}_j > \tau$ where τ is a specific threshold (Tardos uses $\tau = 20c\lceil 1/p_{fa}\rceil$). Note that the functions g_0, g_1 and f_P are such that the expectation of the accusation score of a colluder is maximized while the expectation and the variance of the score of an innocent are both fixed, whatever the colluding strategy. Improvements of computation of the scores and of the threshold have been done in [23]. In [12, 24], the authors define two kinds of decoder to measure the capacity of a fingerprinting scheme given B (resp. B_j) the random variable associated to the symbol at one position in the pirated sequence (resp. in the sequence of the user j):

1. **simple decoder**: the achievable rate R_s for a simple decoder is defined as the mutual information between the pirated sequence forged by a collusion $\mathcal{C}$ and the sequence of a user j (in expectation over P, Eq. (2)):
$$\mathrm{R}_s = \mathbb{E}_P \left[I\left(B; B_j\right) \middle| P \right]. \tag{8}$$

2. **joint decoder**: the achievable rate R_j for a joint decoder is defined as the mutual information between the pirated sequence forged by a collusion $\mathcal{C}$ and the sequences of a collusion $\mathcal{C}'$ with size c' (in expectation over P):
$$\mathrm{R}_j = \frac{1}{c'}\mathbb{E}_P \left[I\left(B; \{B_j\}_{j\in\mathcal{C}'}\right) \middle| P \right]. \tag{9}$$

Tardos accusation functions belong to the simple decoder class: the score is computed for each user. In this article, we are interested in this class (the joint decoder implies best performances but its complexity is important [25]). The computation of achievable rate R_s allows us to measure the efficiency of a collusion attack on Tardos codes.

c	Θ_{WCA}
2	(0. 0.5 1)
3	(0. 0.651 0.349 1.)
4	(0. 0.487 0.5 0.513 1.)
5	(0. 0.594 0. 1. 0.406 1.)

Table 1: Numerical values of WCA attack strategy for collusion of sizes $c = 2, 3, 4, 5$. Note that these values are computed for a simple decoder following the Tardos distribution (Eq. (2)) and does not take into account a possible improvement of the bias distribution as in [10, 26].

3.2 The worst case attack

An attack strategy (or colluding strategy) defines the process used by a collusion $\mathcal{C}$ for forging a fingerprint sequence $\mathbf{b}$ under the marking assumption. It consists in selecting, for each position $i \in [m]$, the symbol of a "candidate" colluder (see Eq. (3)). For each position, the value $\mathbf{b}(i)$ depends on the number of "1" symbols (or "0" by symmetry) that the collusion has at this position. An attack strategy is completely defined by a vector $\Theta = (\Theta(0) \dots \Theta(c)) \in [0, 1]^{c+1}$ in a stochastic way (same methodology as in [8]). We have for each $k \in [c + 1]$:

$$\Theta(k) = \mathbb{P}\left(B = 1 \middle| \sum_{j \in \mathcal{C}} B_j = k \right). \tag{10}$$

We assume that the collusion uses the same strategy for each symbol of the pirated sequence (if a collusion choose to change the strategy at each position, this technique is also a strategy which can be also modeled by a Θ-vector). In order to comply with the marking assumption, we have $\Theta(0) = 0$ and $\Theta(c) = 1$. Always in [8], the authors propose an attack strategy which minimizes the achievable rate for simple decoder, this attack is called "Worst Case Attack" (WCA):

$$\Theta_{WCA} = \arg\min_{\Theta} \mathrm{R}_s(\Theta). \tag{11}$$

This attack minimizes the mutual information between the binary sequence of a member of the collusion and the pirated sequence and Eq. (11) has to be solved using numerical optimization techniques. Table 1 gives examples (from [8]) of Θ_{WCA} for collusion of sizes $c = 2, 3, 4, 5$.

The WCA allows a collusion to forge a pirated sequence which decreases the error exponent of the probability that a member gets accused when $m \to \infty$. However the reader has to note that this attack assumes that the colluders know exactly their symbols (or are able to see difference between a "0" symbol and a "1" symbol in their sequences).

Since watermarking techniques are used to hide sequences in multimedia contents and can prevent colluders to estimate accurately their symbols, the next section tackles the impact of security (linked to symbols estimation) on attack strategies.

4 Impact of watermarking security: a theoretical analysis

4.1 Motivations

Watermarking techniques deal with three important constraints: the imperceptibility, the security and the robustness. We propose to mathematically express the security by the estimation error rate of the symbols by the member of a collusion before the creation of the pirated sequence. On the other hand, the robustness can be expressed by bit error rate between the original pirated sequence and attacked pirated sequence decoded by the distributor before accusation process. Transparency of watermarking embedding techniques guaranties that the content is not degraded by adding a fingerprint sequence. Inherent robustness of watermarking is used to extract the fingerprint sequence when the content suffers common process. We consider in this section these two errors and the consequences on the final accusation.

4.2 Collusion considering security constraint

One way to express the security of the watermarking scheme is the estimation error rate ϵ of symbols by a collusion for each position $i \in [m]$. The more secure the scheme is, the closer to 0.5 the value of ϵ is. Considering this new assumption, a collusion $\mathcal{C}$ correctly decodes their sequences $\{\mathbf{b}_j\}_{j\in\mathcal{C}}$ if $\epsilon = 0$. Moreover, each member will be not able to know if his symbol is the same than the symbol of another member. In this article, we assume that the watermarking scheme respects Kerckhoffs' principle [17]. Then, we can assume that a collusion knows the security level of the scheme and consequently can estimate the error ϵ.

We denote $\hat{\mathbf{b}}_j(i)$ the symbol decoded by the colluder $j \in \mathcal{C}$ at position $i \in [m]$ and $\hat{B}_j$ the associated random variable with property:

$$\mathbb{P}\left(\hat{B}_j = 1 \middle| B_j = 0\right) = \mathbb{P}\left(\hat{B}_j = 0 \middle| B_j = 1\right) = \epsilon. \tag{12}$$

The collusion forges a pirated sequence $\hat{\mathbf{b}} \in \mathbb{F}_2^m$ in the estimated domain using attack strategy Θ now defined by:

$$\Theta(k) = \mathbb{P}\left(\hat{B} = 1 \middle| \sum_{j\in\mathcal{C}} \hat{B}_j = k\right), \tag{13}$$

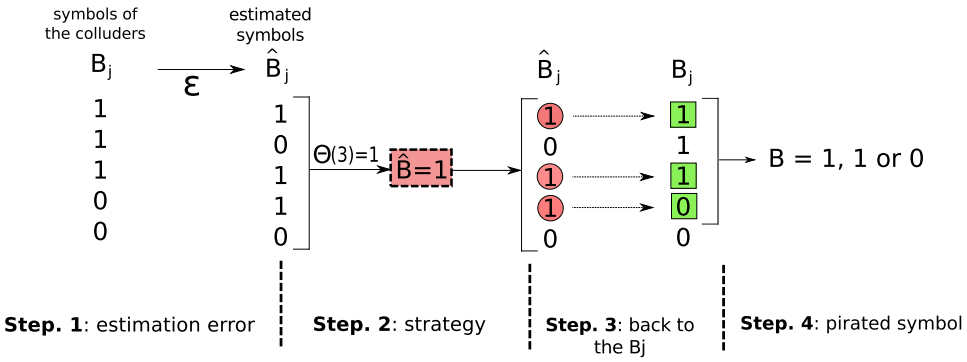


Figure 2: Collusion process for a secure watermarking scheme with $c = 5$ colluders and $\Theta(3) = 1$ (example). **Step. 1:** the colluders decode three "1" symbols $\hat{B}_j$. **Step. 2:** because $\Theta(3) = 1$, the strategy gives $\hat{B} = 1$. **Step. 3:** the coalition looks for the $\hat{B}_j$ which correspond to the $\hat{B}_j = \hat{B} = 1$. **Step. 4:** the pirated symbol B is uniformly chosen among the selected B_j: $\mathbb{P}(B = 1) = 2/3$ and $\mathbb{P}(B = 0) = 1/3$ in this case.

where $\hat{B}$ is the random variable associated to the symbol of foreseen pirated sequence $\hat{\mathbf{b}}$ at a position $i \in [m]$. The true pirated sequence $\mathbf{b}$ is then made according to:

$$\forall i \in [m],\ \mathbf{b}(i) = \mathbf{b}_{j'}(i), \tag{14}$$

with j' chosen in a uniform way in the set:

$$\left\{ j \in \mathcal{C} :\ \hat{\mathbf{b}}_j(i) = \hat{\mathbf{b}}(i) \right\}. \tag{15}$$

Figure 2 illustrates the colluding process for $c = 5$ colluders.

It is important to point out that the embedding process of the sequences is stochastic. For hiding the same symbol, the modification of the content will be not the same for each user. For example, for the spread-spectrum technique Circular Watermarking (CW) [27], a random parameter is used for spreading marked correlations on the whole decoding region. After secret carriers estimation, the decoded symbol consequently suffers from an estimation error ϵ as presented in Figure 3.

Hence, estimation of symbols $\hat{B}_j$ by colluders is not deterministic (in Figure 2, the symbol "1" is estimated as a "0" by colluder $j = 1$ but correctly estimated by colluder $j = 0$). Moreover we can apply this model without considering erasure, since a lot of watermarking schemes (SS, ISS, SCS, etc) do not consider erasures during their decoding stage. Dealing with erasure in fingerprinting is a major problem which receives a lot of concern in the community [5, 28, 29]. Note that the colluding technique we present here complies with the marking assumption thanks to Eq. (14) and Eq. (15) (in

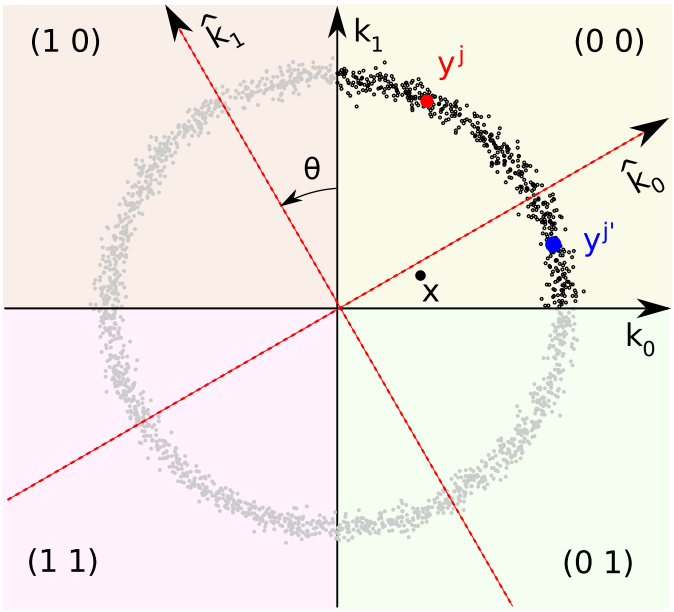


Figure 3: Illustration of watermarking of one host content $\mathbf{x}$ by CW for several users with the same message $\mathbf{b} = (0\ 0)$. Watermarked contents $\{\mathbf{y}^j\}_j$ are spread in the whole decoding region to avoid security fail in WOA context. If we consider two users, watermarked contents $\mathbf{y}^j$ and $\mathbf{y}^{j'}$ are different and located in the codeword corresponding to (0 0). However, with a wrong estimation of the carriers ($\hat{\mathbf{k}}_0$ and $\hat{\mathbf{k}}_1$ instead of $\mathbf{k}_0$ and $\mathbf{k}_1$ with $\widehat{(\mathbf{k}_i, \hat{\mathbf{k}}_i)} = \theta$), user j will properly decode the message (0 0) whereas user j' will decode (0 1). Note that two bits are embedded here for illustration purposes (in our practical analysis in Sec. 5, only one bit is hidden per chunk).

the end the colluder will only output 0 or 1's independently of the chosen strategy).

We now define the achievable rate for simple decoding R_s taking into account the assumption on watermarking security:

$$\begin{aligned}
\mathrm{R}_s(\Theta, \epsilon) &= I(B; B_j|P) \\
&= \mathbb{E}_P[I(B; B_j)|P] \\
&= \mathbb{E}[H(B) - H(B|B_j)|P] \\
&= \mathbb{E}_P[H(B) - (pH(B|B_j = 1) \\
&\quad + (1-p)H(B|B_j = 0))|P = p] \\
&= \mathbb{E}_P[H_b(p_1) - (pH_b(p_2) + (1-p)H_b(p_3))],
\end{aligned} \tag{16}$$

where probabilities p_1, p_2 and p_3 are given by:

$$p_1 = \mathbb{P}(B = 1|P = p), \tag{17}$$

$$p_2 = \mathbb{P}(B = 1|B_j = 1, P = p), \tag{18}$$

$$p_3 = \mathbb{P}(B = 1|B_j = 0, P = p). \tag{19}$$

Computations of p_1, p_2 and p_3 are given in appendix A. Using minimization of the rate by the simplex algorithm [30], we compute the ϵ-Worst Case Attack which minimizes the achievable rate $\mathrm{R}_s(\Theta, \epsilon)$ for some values of ϵ. Results are shown in Table 2.

	$c = 3$	$c = 4$
$\epsilon = 0.$	(0. 0.651 0.349 1.)	(0. 0.487 0.5 0.513 1.)
$\epsilon = 0.05$	(0. 0.726 0.274 1.)	(0. 0.543 0.5 0.457 1.)
$\epsilon = 0.1$	(0. 0.830 0.170 1.)	(0. 0.620 0.5 0.379 1.)
$\epsilon = 0.15$	(0. 0.982 0.018 1.)	(0. 0.734 0.5 0.266 1.)
$\epsilon = 0.2$	(0. 1. 0. 1.)	(0. 0.908 0.5 0.091 1.)
$\epsilon > 0.2$	(0. 1. 0. 1.)	(0. 1. 0.5 0. 1.)

Table 2: Numerical values of $\theta_{\epsilon\text{-WCA}}$ functions of ϵ for $c = 3, 4$. For $c = 2$, for all $\epsilon \in [0, 0.5]$, $\theta_{\epsilon\text{-WCA}} = (0.\ 0.5\ 1.)$.

We now compare in term of achievable rates the ϵ-WCA with the classical WCA and other strategies [9] like:

- Interleaving attack, this attack consists in selecting one symbol of a colluder in a random way:

$$\forall k \in [c+1],\ \Theta_{\text{interleaving}}(k) = k/c. \tag{20}$$

- Majority Vote, this attack consists in selecting the majority symbol among the collusion:

$$\forall k \in [c+1],\ \Theta_{\text{MAJ}}(k) = \begin{cases} 0 \text{ if } k \in [\![0, c/2[\![, \\ 1/2 \text{ if } k = c/2, \\ 1 \text{ if } k \in]\!]c/2, c]\!]. \end{cases} \tag{21}$$

As we will see in section 5, this attack can be equivalent to an averaging attack.

- Minority vote, this attack consists in selecting the minority symbol among the collusion:

$$\forall k \in [c+1],\ \theta_{\text{MIN}}(k) = \begin{cases} 0 \text{ if } k = 0, \\ 1 \text{ if } k \in]\!]0, c/2[\![, \\ 1/2 \text{ if } k = c/2, \\ 0 \text{ if } k \in]\!]c/2, c[\![, \\ 1 \text{ if } k = c. \end{cases} \tag{22}$$

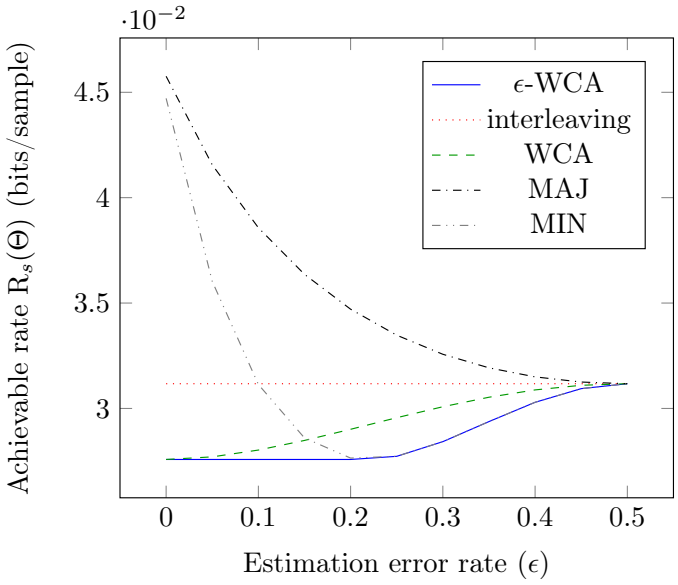


Figure 4: Values of $R_s(\Theta)$ (Θ = WCA, ϵ-WCA or interleaving) w.r.t the estimation error rate ϵ for $c = 4$. We can see that ϵ-WCA is able to decrease the accusation performance of the colluders.

Figure 4 shows values of achievable rate as a function of the estimation error rate ϵ for collusions of size $c = 4$ using interleaving, MAJ, MIN, WCA and ϵ-WCA strategies.

This figure enables to list different observations. (1) ϵ-WCA is the best way for colluders to decrease their accusation rate. Note that when $\epsilon = 0$ (perfect estimation) and $\epsilon = 0.5$ (collusion does not know what it decodes) ϵ-WCA is equivalent to WCA. (2) The interleaving attack does not depend on the estimation error rate (because colluders do not use knowledge of ϵ to forge the sequence). (3) The majority vote is the worst strategy for the colluders (the rate achieved by ϵ-WCA represents 60% of the one achieved by majority vote). (4) All the strategies are equivalent when ϵ achieves its maximum (0.5). (5) In this setup ϵ-WCA tends to a minority vote strategy when $\epsilon > 0.2$ (this is also confirmed by Table 2).

4.3 Attack after colluding strategy

After a colluding attack, the forged sequence is embedded into the pirated content. Before its possible diffusion on public networks, the content may be damaged by robustness attacks such as compression or noise addition. It means that the distributor decodes a symbol $\mathbf{b}'(i)$ ($i \in [m]$) instead of the symbol $\mathbf{b}(i)$ with bit error rate η modeling a binary symmetric channel. The corresponding random variable B' follows:

$$\mathbb{P}(B' = 1|B = 0) = \mathbb{P}(B' = 0|B = 1) = \eta. \tag{23}$$

We compare here the achievable rates of insecure embedding schemes ($\epsilon = 0$) with respect to secure embedding schemes ($\epsilon > 0$) including a memoryless attack channel (modeled by a BSC) with characteristic η.

Note that BSC is a general model which can simulate different attacks at the decoding stage (averaging, removal attack [31], etc). The memoryless assumption is the most general, and it is indeed realistic for practical contents such as images of videos when the size of each chunk is long enough (i.e. the host can be considered as i.i.d from one chuck to another), for stochastic embedding (i.e. when the robustness is different from one content to the other), and for embeddings using different keys from one chunk to another (including a possible repetition of these keys). We now compute an updated version of the achievable rate R'_s:

$$\mathrm{R}'_s(\Theta, \epsilon, \eta) = \mathbb{E}_P[I(B'; B_j)|P]. \tag{24}$$

This rate is computed using the same technique as for Eq. (16) with:

$$p'_1 = \mathbb{P}(B = 1|P = p) = (1 - \eta)p_1 + \eta(1 - p_1), \tag{25}$$

$$p'_2 = \mathbb{P}(B = 1|B_j = 1, P = p) = (1 - \eta)p_2 + \eta(1 - p_2), \tag{26}$$

$$p'_3 = \mathbb{P}(B = 1|B_j = 0, P = p) = (1 - \eta)p_3 + \eta(1 - p_3). \tag{27}$$

We compute the achievable rate for the ϵ-WCA taking into account the estimation error rate ϵ and the decoding error rate η (unknown by both colluders and distributor).

Figure 5 shows the difference Δ_{rate} between achievable rates for secure embedding schemes: $\mathrm{R}'_s(\epsilon - WCA, \epsilon, \eta)$ and insecure ones: $\mathrm{R}'_s(WCA, 0, \eta)$ for $c = 4$ colluders. As can be seen, the difference between the two schemes is significant for secure schemes (ϵ close to 0.5) with a weak attack after the colluding strategy (η close to 0).

In Figure 6, for $c = 4$ colluders and given ϵ, we find the BER η_1 such as the achievable rate R'_s after ϵ-WCA is the same for insecure schemes ($\epsilon = 0$) given a BER η_2. η_1 is tantamount to the maximum probability of error that has to handle the insecure schemes in order to offer the same achievable rate. Formally, we look for the root η_1 which satisfies:

$$\mathrm{R}'_s(\Theta_{\epsilon-WCA}, \epsilon, \eta_1) = \mathrm{R}'_s(\Theta_{WCA}, 0, \eta_2), \tag{28}$$

when η_1 is computed using the Brent-Dekker algorithm [32].

This figure enables to quantify the tradeoff between robust schemes (given by η_2) and secure schemes (given by η_1). When ϵ grows up, a secure

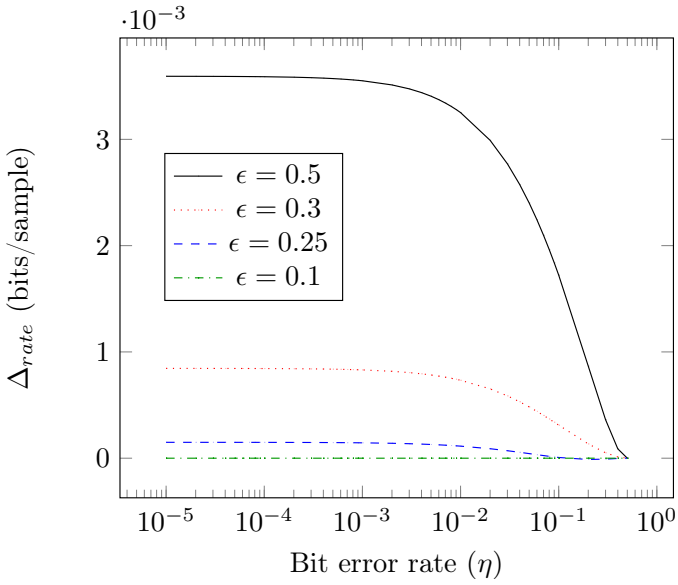


Figure 5: The gap between robustness attack and security combined with robustness attack is represented by $\Delta_{rate} = \mathrm{R}'_s(\Theta_{\epsilon-WCA}, \epsilon, \eta) - \mathrm{R}'_s(\Theta_{WCA}, 0, \eta)$ w.r.t. bit error rate η for $\epsilon = 0.5, 0.3, 0.25, 0.1$, $c = 4$.

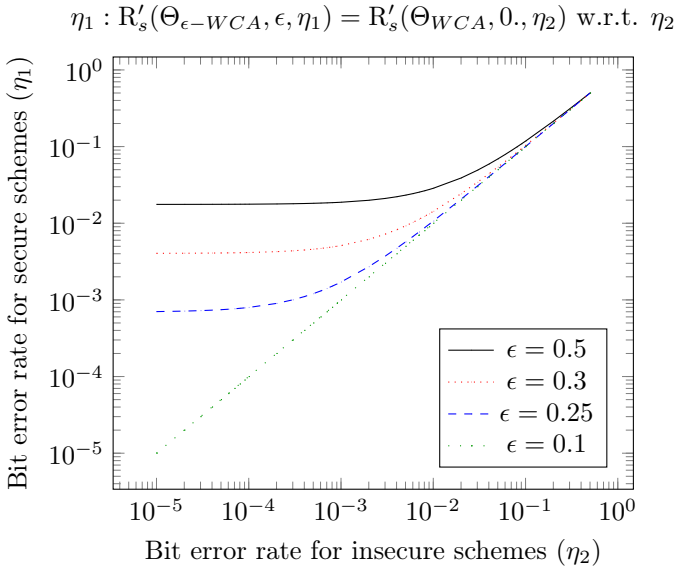


Figure 6: The difference in term of BER between insecure schemes and secure schemes for the same accusation rate is measured by the bit error rate η_1 (secure schemes) functions of bit error rate η_2 (insecure schemes) for $\epsilon = 0.5, 0.3, 0.25, 0.1$, and $c = 4$ colluders.

watermarking scheme will be more prone to handle errors than an insecure watermarking scheme. For the same mutual information between the decoded pirated sequence and the sequence of a colluder, a BER η_2 with value $1.e-05$ for an insecure watermarking scheme corresponds to a totally secure embedding scheme ($\epsilon = 0.5$) with a BER $\eta_1 = 1.761e-02$. Note however that the difference between secure and insecure scheme becomes negligible for large BER. We evaluate the impact of security and robustness in the following section using a practical example of watermarking scheme.

5 Impact of watermarking security: a practical analysis on stochastic spread-spectrum

5.1 Stochastic spread-spectrum for fingerprinting

In this section we complete results from the previous section with a practical watermarking scheme using spread-spectrum modulation [33] for Tardos based fingerprinting.

In our model depicted in Figure 7, the multimedia content of size T that a distributor watermarks is first divided (in the spatial or transform domain) into N_c chunks $\{\mathbf{x}_k\}_{k\in N_c}$ with length N_v: $N_c = T/N_v$. We assume that each chunk is Gaussian distributed. The message to be hidden in the content is a Tardos binary sequence of length m. According to a secret key, the distributor also generates m secret carriers $\{\mathbf{k}_i\}_{i\in[m]}$ with length N_v and unitary norm ($\|\mathbf{k}_i\|_2 = 1$). Each carrier hides one binary symbol into one chunk. In order to spread all the information into the content and to increase the robustness, the embedding of the Tardos code is repeated for the m following chunks and so on until the end of the document. Then, the number of repetition of the whole Tardos code is $N_r = \lfloor T/(m \times N_v)\rfloor = \lfloor N_c/m\rfloor$.

In order to increase the security of the watermarking scheme, we use here a new stochastic version of the classical spread-spectrum modulation named Stochastic Spread-Spectrum (SSS). This method is similar to classical Spread-Spectrum embedding, except that for each chunk, we add noise in a direction $\mathbf{s}_k$ randomly chosen in the orthogonal subspace of the secret carrier $\mathbf{k}_i$. Note that this trick is similar to the security measure proposed by Cao and Huang [34] to increase the security of Circular Watermarking.

Formally, for each $k \in N_c$, we consider the Gaussian host chunk $\mathbf{x}_k$ and its associated secret carrier $\mathbf{k}_i$ ($i = k \bmod m$). The watermarked chunk for user $j \in [n]$ is denoted by $\mathbf{y}_k^j$, the watermark signal by $\mathbf{w}_k^j$ and the added random noise by $\mathbf{s}_k^j$. These signals belong to $\mathbb{R}^{N_v}$. The embedding of the binary symbol $b_j \in \mathbb{F}_2$ into $\mathbf{x}_k$ follows:

$$\mathbf{y}_k^j = \mathbf{x}_k + \mathbf{w}_k^j = \mathbf{x}_k + \alpha(-1)^{b_j}\mathbf{k}_i + \gamma\mathbf{s}_k^j, \tag{29}$$

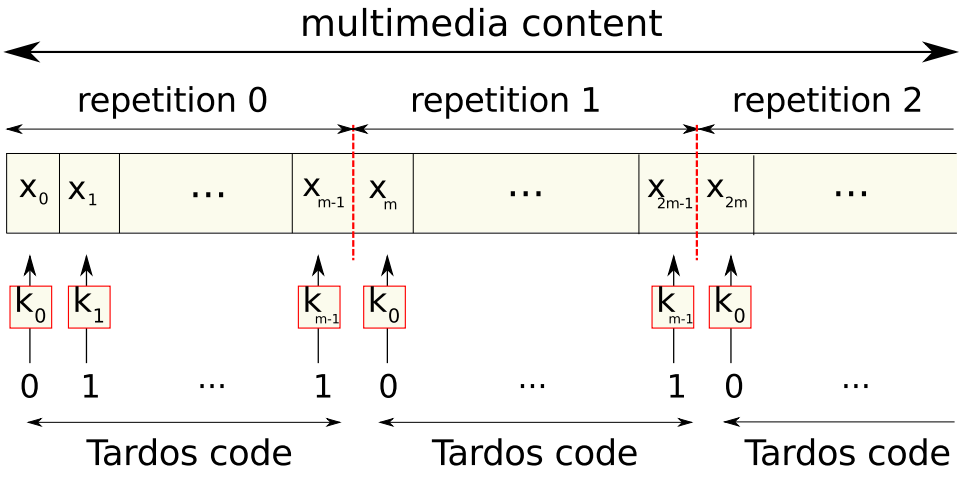


Figure 7: Embedding process for Tardos binary sequence into multimedia content. The Tardos code is repeated on the whole content. Each symbol of the sequence is embedded into one chunk using classical spread-spectrum with one secret carrier.

with α a scalar setting the distortion and γ a scalar setting the strength of the spread of the signal in a orthogonal direction of the secret carrier.

Signal $\mathbf{s}_j^k$ creates a watermarked chunk different for each user $j \in [n]$. This property enables a noisy estimation of symbols per colluder. Note that the distributor could design strategies to exploit the signal $\mathbf{s}_j^k$ to identify users. Each signal being independent for each user, classical spread spectrum zero-bit watermarking techniques could be used to help identification (this is the idea of independent fingerprinting developed in [35]). However, in order to avoid any potential security attack associated with this side information, we choose to generate a purely random signal $\mathbf{s}_j^k$ and not to use it during the decoding side. This way all the security issues remain associated with the embedding of the Tardos code. SSS thus complies with assumptions of the ϵ-WCA (Section 4.2). Moreover, this technique does not modify classical SS decoding rules because we do not modify the subspace spanned by the secret carriers. So, for the following derivations, we do not consider the parameter γ in the sequel.

We have $\sqrt{\alpha^2+\gamma^2} = \sigma_x\sqrt{N_v}10^{WCR/20}$, where the Watermark-to-Content power Ratio is:

$$\mathrm{WCR} = 10\log_{10}\left(\frac{\sigma_{\mathbf{w}}^2}{\sigma_{\mathbf{x}}^2}\right). \tag{30}$$

For a better comprehension of the following equations, we will only focus on distortion induced by $\mathbf{k}_i$ using $\mathrm{WCR}_\alpha = 10\log_{10}\left(\frac{\alpha^2\sigma_{\mathbf{k}_i}^2}{\sigma_{\mathbf{x}}^2}\right)$.

Decoding is performed using correlations between watermarked chunks

and secret carriers:

$$b'_j = \begin{cases} 0 & \text{if } \left\langle \mathbf{y}_k^j \middle| \mathbf{k}_i \right\rangle \geq 0, \\ 1 & \text{if } \left\langle \mathbf{y}_k^j \middle| \mathbf{k}_i \right\rangle < 0, \end{cases} \tag{31}$$

where b'_j is the estimated symbol.

5.2 Practical ϵ-WCA

Each member of a collusion $j \in \mathcal{C}$ owns N_c chunks $\left\{\mathbf{y}_k^j\right\}_{k \in N_c}$ with length N_v. The first natural attack is an averaging attack for each chunk:

$$\forall k \in N_c,\ \mathbf{y}_k = \frac{1}{c} \sum_{j \in \mathcal{C}} \mathbf{y}_k^j. \tag{32}$$

For antipodal spread-spectrum techniques, this attack is equivalent to a Majority Vote colluding strategy when colluders perfectly decode their symbols ($\epsilon = 0$). However, we can see on Figure 4 that the achievable rate for this strategy is always below the ones of the other strategies. Colluders have consequently no interest in the averaging attack and should use the ϵ-WCA instead.

Because the ϵ-WCA is the best collusion strategy, their first step consists in estimating error rate ϵ. Note that distributor and adversaries only know a lower bound of the estimation error rate ϵ. However, this knowledge is sufficient to produce a collusion attack which is better than the classical WCA. Moreover, if we consider all the implications of the Kerckhoffs principle, the colluders can accurately estimate at home the parameter ϵ since it depends only of public parameters (the dimension of feature vectors, the number of observations, the distortion and a model of the host signal).

In order to simplify the following computations, we assume that $b_j = 0$. According to Kerckhoffs' principle, the only unknown parameters for a colluder $j \in \mathcal{C}$ are the m secret carriers. However, he knows that the sequence is repeated and that the carrier and the symbol associated to the chunk $\mathbf{y}_i^j$ are the same for each chunk $\mathbf{y}_{i+lm}^j$ with $l \in [N_r]$ (we are here in the Constant Message Attack [36] framework). Thanks to this information, he can compute $\hat{\mathbf{k}}_i$, an estimation of the secret carrier $\mathbf{k}_i$ ($i \in [m]$) by averaging on the number of repetitions of the Tardos code the watermarked chunks $\mathbf{y}_k^j$ which corresponds to the same symbol ("0" in our example) and the same carrier $\mathbf{k}_i$. Formally:

$$\hat{\mathbf{k}}_i = \frac{1}{\alpha N_r} \sum_{l \in [N_r]} \mathbf{y}_{i+lm}^j. \tag{33}$$

The bigger the number of repetitions, the more precise the estimation. Note that if several colluders try together to estimate $\mathbf{k}_i$ because they own

several chunks watermarked with the same secret key but with different symbols, they can combine this technique with source separation [19].

Figure 8 shows the bit error rate (estimation error rate ϵ) between original symbols and symbols obtained using estimated carriers constructed in Eq. (33) with respect to the number of repetitions N_r. Results are computed in expectation over $m = 10,000$ secret carriers and $N_c = 10,000$ chunks with size $N_v = 256$. As can be seen, we obtain a better estimation when the numbers of repetitions increases. With this estimation error rate, the whole collusion $\mathcal{C}$ can forge a pirated content by mixing their chunks according to the ϵ-WCA (for each repetition of the Tardos code).

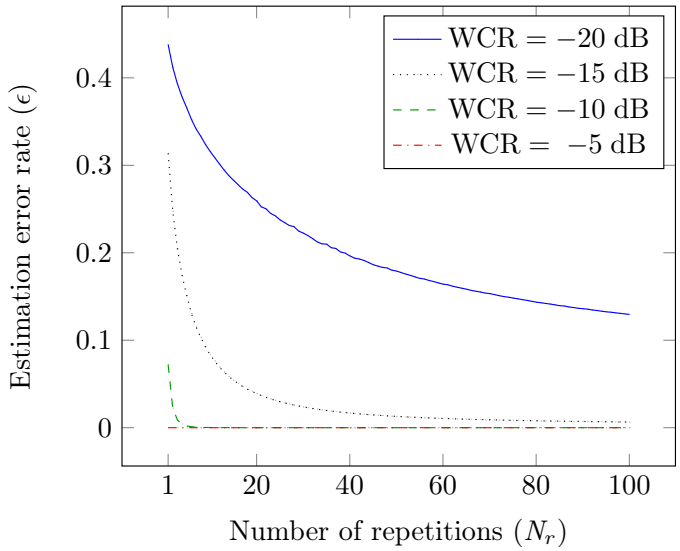


Figure 8: Impact of the security attack: Estimation error rate ϵ between original symbols and symbols obtained using estimated carriers constructed in Eq. (33) w.r.t the number of repetitions N_r. Results are computed in expectation over $m = 10,000$ secret carriers and $N_c = 10,000$ chunks with size $N_v = 256$.

In order to compute the impact of the estimation error on a security attack, a colluder j can also compute ϵ with respect to the angular deviation between the true and the estimated keys. Indeed, ϵ is function of the angle θ between the two unitary vectors $\hat{\mathbf{k}}_i$ and $\mathbf{k}_i$. Given $\left\langle \mathbf{k}_i \middle| \hat{\mathbf{k}}_i \right\rangle = \cos\theta$, the correlations between watermarked chunks and estimated carriers follow:

$$\left\langle \mathbf{y}_k^j \middle| \hat{\mathbf{k}}_i \right\rangle = \left\langle \mathbf{x}_k + \alpha \mathbf{k}_i \middle| \hat{\mathbf{k}}_i \right\rangle = \left\langle \mathbf{x}_k \middle| \hat{\mathbf{k}}_i \right\rangle + \alpha \cos\theta. \tag{34}$$

We have $\left\langle \mathbf{x}_k \middle| \hat{\mathbf{k}}_i \right\rangle \sim \mathcal{N}(0, \sigma^2_{\mathbf{x}_k})$ and $\left\langle \mathbf{y}_k^j \middle| \hat{\mathbf{k}}_i \right\rangle \sim \mathcal{N}(\alpha\cos\theta, \sigma^2_{\mathbf{x}_k})$. ϵ is then:

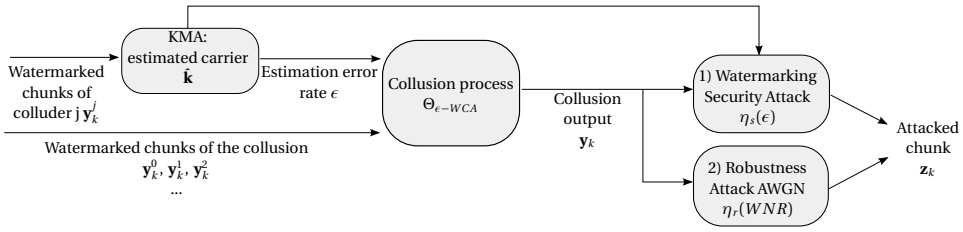


Figure 9: Attack process: the estimation error rate is obtained after secret carrier estimation to apply the ϵ-WCA colluding strategy. After that, the pirated chunks can be attacked using a security attack (using ϵ) or a robustness attack.

$$\begin{aligned}\epsilon &= \mathbb{P}(B = 1|\hat{B} = 0) \\ &= \frac{1}{2}\left(1 + \text{erf}\left(\frac{-\alpha\cos\theta}{\sqrt{2}\sigma_x}\right)\right) \\ &= \frac{1}{2}\left(1 + \text{erf}\left(\frac{-\sqrt{N_v}10^{\frac{\text{WCR}_\alpha}{20}}\cos\theta}{\sqrt{2}}\right)\right).\end{aligned} \tag{35}$$

and the relation that gives θ w.r.t ϵ is:

$$\cos\theta = -10^{\frac{-\text{WCR}_\alpha}{20}}\sqrt{2}\frac{\text{erf}^{-1}(2\epsilon - 1)}{\sqrt{N_v}}. \tag{36}$$

5.3 Attacks after colluding strategy

The collusion now performs the ϵ-WCA fingerprinting attack and forged a pirated sequence with N_c chunks $\{\mathbf{y}_k\}_{k\in N_c}$ by mixing their chunks (see Eq. (3)). The adversary can now also assess directly the robustness or the security of the watermarking scheme. The content can be damaged (in a intentional way or not) and will produce errors during the decoding step by the distributor. In this section, we compare two attacks:

- a security **removal attack** taking into account the estimated carriers $\hat{\mathbf{k}}_i$,
- a simple **AWGN attack**.

Figure 9 illustrates the whole attack process (before and after colluding strategy).

5.3.1 Removal attack

Without loss of generality we assume that $b_j = 0$, the collusion produces for each $k \in N_c$ an attacked chunk $\mathbf{z}_k$:

$$\mathbf{z}_k = \mathbf{y}_k - \alpha\hat{\mathbf{k}}_i = \mathbf{x}_k + \alpha\mathbf{k}_i - \alpha\hat{\mathbf{k}}_i. \tag{37}$$

Because the correlations $\langle \mathbf{z}_k \, | \mathbf{u}_i \rangle$ are distributed according to $\mathcal{N}(\alpha(1 - \cos\theta), \sigma_{\mathbf{x}}^2)$, the bit error rate η_s after this security attack is then:

$$\eta_s = \mathbb{P}(B' = 1 | B = 0) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{-\alpha(1-\cos\theta)}{\sqrt{2\sigma_x^2}}\right)\right). \tag{38}$$

With Eq. (36) and Eq. (38), we find the relation between ϵ and η_s:

$$\eta_s(\epsilon) = \frac{1}{2}\left(1 + \mathrm{erf}\left(-\sqrt{\frac{N_v}{2}} \times 10^{\mathrm{WCR}_\alpha/20} - \mathrm{erf}^{-1}(2\epsilon - 1)\right)\right). \tag{39}$$

Note that $\eta_s(\epsilon)$ is decreasing functions of ϵ. We are now able to compute achievable rates $\mathrm{R}'(\epsilon - WCA, \epsilon, \eta_s(\epsilon))$ with $N_v = 256$ and different values for WCR. Results are shown in Figure 10. Curves of $\eta_s(\epsilon)$ with $N_v = 256$ and several WCR are shown too in Figure 11.

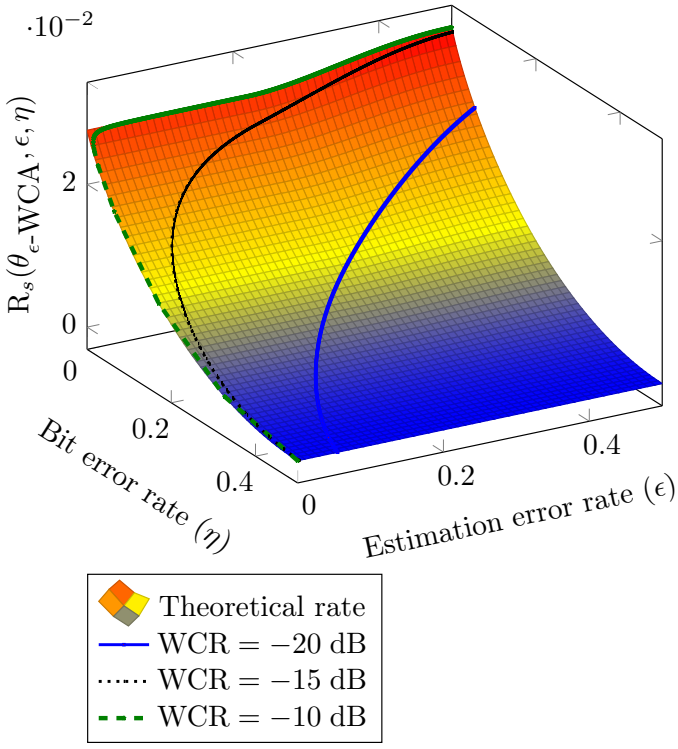


Figure 10: Achievable rates for the security attack proposed in (37).

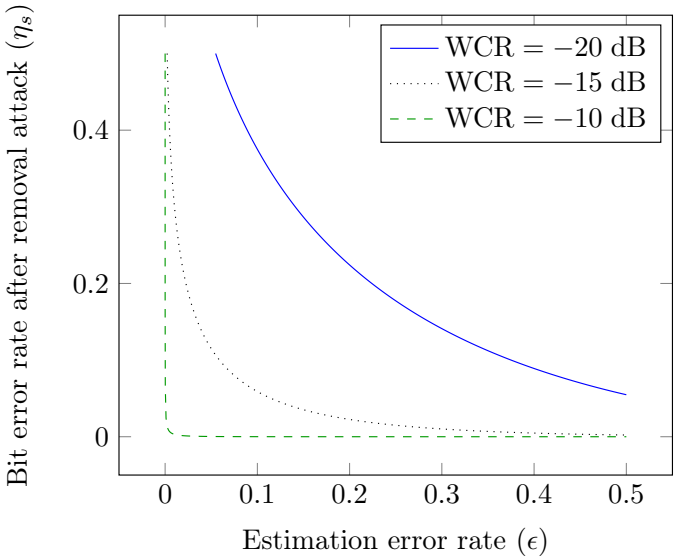


Figure 11: Bit error rate η_s w.r.t. ϵ for chunks with size $N_v = 256$. We note that when ϵ is increasing (colluder do not know exactly their embedded symbols), value of bit error rate is decreasing (the attack is also less powerful).

5.3.2 Robustness attack

We now consider an AWGN attack with a Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, 1/N_v)$. The channel output becomes:

$$\mathbf{z}_k = \mathbf{y}_k + \beta\mathbf{n} = \mathbf{x}_k + \alpha\mathbf{k} + \beta\mathbf{n},$$

where β depends on the Watermark-to-Noise Ratio (WNR): $\beta = \alpha 10^{-\mathrm{WNR}/20}$.

Correlations $\langle \mathbf{z}_k \, | \mathbf{n} \rangle$ are now distributed according to $\mathcal{N}\left(\alpha, \sigma_{\mathbf{x}}^2 + \frac{\beta^2}{N_v}\right)$. We obtain the bit error rate:

$$\begin{aligned} \eta_r &= \mathbb{P}(M' = 1 | M = 0) \\ &= \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{-\alpha}{\sqrt{2\left(\sigma_x^2 + \beta^2/N_v\right)}}\right)\right) \\ &= \frac{1}{2}\left(1 + \mathrm{erf}\left(-\sqrt{\frac{N_v}{2}}\frac{1}{\sqrt{10^{-\mathrm{WCR}_\alpha/10} + 10^{-\mathrm{WNR}/10}}}\right)\right). \end{aligned} \tag{40}$$

5.3.3 Comparison between the two attacks

We solve the equation $\eta_s(\epsilon_p) = \eta_r(\mathrm{WNR}_{max})$ in order to find the pivot value ϵ_p for with the robustness attack is more/less efficient than the security attack. In fact, for all $\epsilon < \epsilon_p$, $\eta_s(\epsilon) > \eta_s(\epsilon_p) = \eta_r(\mathrm{WNR}_{max})$. Since the

goal of the collusion is to increase the bit error rate while the distributor decodes the pirated sequence (in order to minimize the achievable rate, see Figure 10), for a given WNR_{max}, colluders should perform a security attack when $\epsilon < \epsilon_p$ instead of a robustness attack with $\text{WNR} < \text{WNR}_{max}$. On the other hand, the goal of the distributor is to avoid potential security attack and consequently to have a watermarking scheme with security $\epsilon > \epsilon_p$. To solve this equation, we use the inverse function of Eq. (39). For $\eta_s = \eta_r$, we obtain:

$$\epsilon_p(\text{WNR}) = \frac{1}{2}\left(1 + \text{erf}\left(\sqrt{\frac{N_v}{2}}\left(-10^{\text{WCR}_\alpha/20} + \frac{1}{\sqrt{10^{-\text{WCR}_\alpha/10} + 10^{-\text{WNR}/10}}}\right)\right)\right). \tag{41}$$

The relation between the security parameter ϵ_p and the WNR of a AWGN attack is shown on Figure 12 for size of chunks $N_v = 256$, and different WNRs. Because the WNR of the removal attack is 0 dB, the security attack is compared with the robustness attack for WNR $\leq$ 0 dB, i.e. when the distortion of the robustness attack is smaller than that of the security attack. This configuration is possible for all ϵ less than $\epsilon_p = 4.97e-01$ (WCR= -20 dB), $4.82e-01$ (WCR= -15 dB), $4.07e-01$ (WCR= -10 dB), $1.24e-01$ (WCR= -5 dB), $1.39e-06$ (WCR= 0 dB). Practically, this means that the colluders prefer the proposed security attack to the AWGN attack inducing the same distortion whenever the embedding scheme has a security flaw such that $\epsilon < \epsilon_p$. On the contrary, we show that for high WNR or low WCR, a distributor should adopt secure embedding schemes with $\epsilon > \epsilon_p$. We can conclude that the security attack requires a tight estimation of the secret key for large WCR and a rough estimation for low WCR. Note however that, according to Figure 8, a large WCR also implies an accurate estimation of the secret key even with few observed contents: one observation ($N_r = 1$) always gives an estimation which is accurate enough to perform the security attack for $N_v = 256$.

5.4 Practical computation of achievable rates

At this level of our study, it is important to note that the marking assumption can always be violated for the removal or AWGN attack since during the decoding stage the modifications of the watermarked chunks can potentially change one symbol to any other random one. This is not specific to our embedding scheme but true for any scheme and the original Tardos accusation process was not optimized for these attacks since they do not respect the marking assumption. Note that the AWGN attack has been specifically considered in [37] with a relaxation of the marking condition and a specific decoder. In this section, we only analyze the impacts of these attacks from

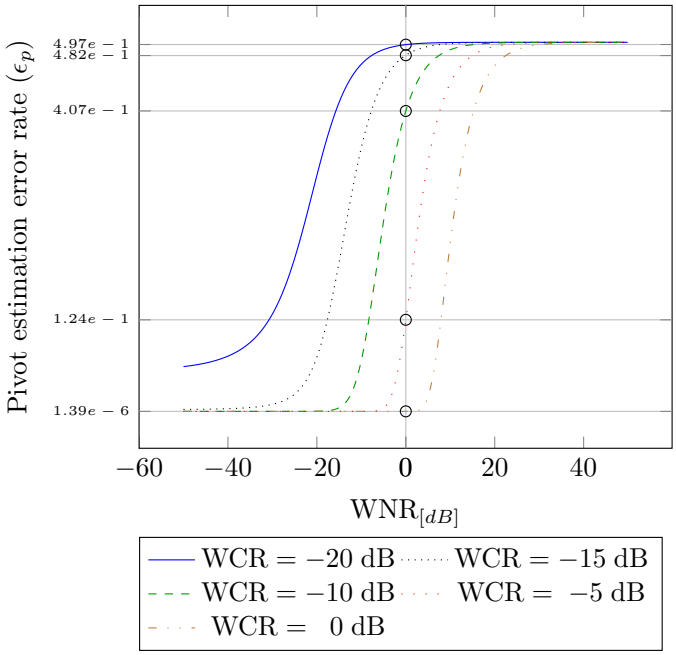


Figure 12: Pivot estimation error rate ϵ_p for which, for $\epsilon < \epsilon_p$, the security attack is more efficient than the AWGN attack w.r.t. the WNR of the AWGN ($N_v = 256$). WNR = 0 dB represents the distortion of the removing attacks.

information theory point of view using achievable rates. Note also that the complexity and the resistance against other possible attacks are currently the two major problems that a practical accusation algorithm has to deal with. Working with information theory measures gives us the advantage to be independent of the accusation strategy.

We compute achievable rates taking account 1) the collusion strategy and 2) a robustness or security attack on this content. We assume that the distributor knows neither the strategy done by colluders nor the attack that the pirated content has suffered. Achievable rates are then computed by the distributor in a practical way in expectation over the number of repetitions N_r of the Tardos sequence and over the values of p_i of this sequence.

Practical values of p_1, p_2, p_3 and then mutual informations are estimated using (16) for each position $i \in [m]$ of the Tardos code using Monte-Carlo process on $N_r = 100,000$ repetitions. Achievable rates are then computed as the mean of the mutual informations on the $m = 10,000$ positions. Figure 13 shows the achievable rates functions of ϵ for the five attack sets: collusion attack, security attack (removal attack), robustness attack (AWGN with WNR= 0 dB), collusion attack + security attack and collusion attack + robustness attack for WCR= −20 dB (a) and WCR= −15 dB (b). This illustrative experiment confirms the fact that a collusion attack followed by

a security attack is the best way for colluders to decrease their accusation rate. As expected, according to Figure 12, the removal attack is more efficient than the AWGN attack when $\epsilon < \epsilon_p$. For example, for WCR= -15 dB, $\epsilon_p = 4.07e{-}01$ (see Figure 12) and the practical achievable rate after security attack is lower than for robustness attack with WNR= 0 dB when $\epsilon < \epsilon_p$.

The general conclusion of this practical analysis, which can be extended to other watermarking schemes, is twofold:

1. the gap between the naive robustness attack and the combination of a fingerprint coalition attack with a watermarking security attack highlights the potential gain of an advised adversary,

2. on the other hand, this gain is minimized to the gain of an interleaving attack if the distributor uses a secure watermarking scheme ($\epsilon = 0.5$).

6 Conclusion

This article highlights the double role of watermarking security, i.e. the possibility to estimate partly or fully the embedding key and the embedded symbols, in the context of watermarking-based fingerprinting. We have first shown that secure schemes prevent the colluders from performing a Worst Case Attack and force them into an interleaving attack which decreases the accusation rate. Secondly, the robustness analysis shows that a watermarking security attack, here a removing attack, applied on spread-spectrum schemes can be more efficient than an AWGN attack even for a very low amount of observations. These two observations motivate the use of secure embedding for the deployment of fingerprinting techniques.

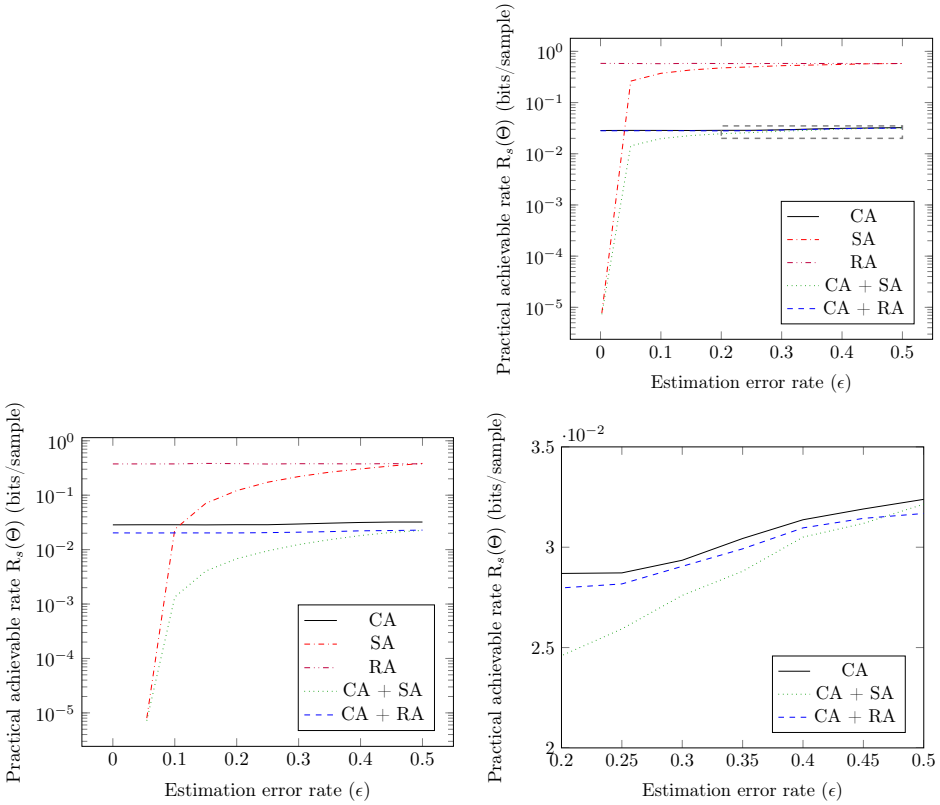
A Computation of achievable rate for simple decoding R_s taking into account the assumption of watermarking security

Using the following notations:

- $\sum_{j\in\mathcal{C}} B_j = \Sigma_B$,
- $\sum_{j\in\mathcal{C}} \hat{B}_j = \Sigma_{\hat{B}}$,

we now compute the expressions of p_1 (17), p_2 (18) and p_3 (19). We have:

$$\begin{aligned} p_1 = \sum_{l=0}^{c}\sum_{k=0}^{c} & \mathbb{P}\left(\Sigma_B = l, \Sigma_{\hat{B}} = k\right) \\ & \times \mathbb{P}\left(B = 1 | \Sigma_B = l, \Sigma_{\hat{B}} = k\right). \end{aligned} \tag{42}$$



(a) WCR=−20 dB (b) WCR=−15 dB

Figure 13: Practical achievable rates functions of estimation error rate ϵ with WCR = a) -20 dB, b) -15 dB. Length of Tardos code: $m = 10,000$. Collusion size: $c = 4$. Number of trials to estimate p_1, p_2 and p_3: $N_r = 100,000$. Length of chunk: $N_v = 256$. CA, SA and RA respectively stand for Coalition Attack, Security Attack using the removal attack and Robustness attack using the AWGN channel with WNR = 0 dB. For WCR = -15 dB, we show too a zoom of the gray rectangle on the main plot.

We introduce the random variable V, which corresponds to the number of $B_j = 1$ which have been decoded to $\hat{B}_j = 0$:

$$V = \#\{j \in \mathcal{C} : B_j = 1, \hat{B}_j = 0\}. \tag{43}$$

For $l, k \in [c+1]$, V gets its values in the set:

$$\Omega = \{i \in \mathbb{N} : i \leq l; i \leq c-k; i \geq l-k\}. \tag{44}$$

Next,

$$\begin{aligned} p_1 = \sum_{l=0}^{c} \Bigg(& \mathbb{P}(\Sigma_B = l) \sum_{k=0}^{c} \Big(\mathbb{P}(\Sigma_{\hat{B}} = k | \Sigma_B = l) \\ & \times \sum_{i \in \Omega} \big(\mathbb{P}(B = 1 | \, V = i, \Sigma_B = l, \Sigma_{\hat{B}} = k) \\ & \times \mathbb{P}(\, V = i | \Sigma_B = l, \Sigma_{\hat{B}} = k) \big) \Big) \Bigg), \end{aligned} \tag{45}$$

using (combinatorial analysis and conditional probabilities):

$$\mathbb{P}(\Sigma_B = l) = \binom{c}{l} p^l (1-p)^{c-l}, \tag{46}$$

$$\begin{aligned} & \mathbb{P}(\Sigma_{\hat{B}} = k | \Sigma_B = l) \\ & = \sum_{i \in \Omega} \binom{l}{i} \binom{c-l}{k-l+i} \epsilon^i (1-\epsilon)^{l-i} \epsilon^{k-l+i} (1-\epsilon)^{c-k-i}, \end{aligned} \tag{47}$$

$$\begin{aligned} & \mathbb{P}(B = 1 | \, V = i, \Sigma_B = l, \Sigma_{\hat{B}} = k) \\ & = \Theta(k) \frac{l-i}{k} + (1 - \Theta(k)) \frac{i}{c-k}, \end{aligned} \tag{48}$$

$$\begin{aligned} & \mathbb{P}(\, V = i | \Sigma_B = l, \Sigma_{\hat{B}} = k) \\ & = \frac{\binom{l}{i} \binom{c-l}{k-l+i} \epsilon^i (1-\epsilon)^{l-i} \epsilon^{k-l+i} (1-\epsilon)^{c-k-i}}{\sum_{t \in \Omega} \binom{l}{t} \binom{c-l}{k-l+t} \epsilon^t (1-\epsilon)^{l-t} \epsilon^{k-l+t} (1-\epsilon)^{c-k-t}}. \end{aligned} \tag{49}$$

We look now for $p_2 = \mathbb{P}_1(B = 1)$ and $p_3 = \mathbb{P}_0(B = 1)$ with $\mathbb{P}_1(.) \equiv \mathbb{P}(.|B_j = 1)$ and $\mathbb{P}_0(.) \equiv \mathbb{P}(.|B_j = 0)$. Using the same technique as for the computation of p_1, we obtain:

$$\begin{aligned} p_2 = \sum_{l=1}^{c} \Bigg(& \mathbb{P}_1(\Sigma_B = l) \sum_{k=0}^{c} \Big(\mathbb{P}(\Sigma_{\hat{B}} = k | \Sigma_B = l) \\ & \times \sum_{i \in \Omega} \big(\mathbb{P}(B = 1 | \, V = i, \Sigma_B = l, \Sigma_{\hat{B}} = k) \\ & \times \mathbb{P}(\, V = i | \Sigma_B = l, \Sigma_{\hat{B}} = k) \big) \Big) \Bigg), \end{aligned} \tag{50}$$

$$p_3 = \sum_{l=0}^{c-1}\Bigg(\mathbb{P}_0(\Sigma_B = l)\sum_{k=0}^{c}\Big(\mathbb{P}(\Sigma_{\hat{B}} = k|\Sigma_B = l) \times \sum_{i\in\Omega}\big(\mathbb{P}(B = 1|\, V = i, \Sigma_B = l, \Sigma_{\hat{B}} = k) \times \mathbb{P}(\, V = i|\Sigma_B = l, \Sigma_{\hat{B}} = k)\big)\Big)\Bigg), \tag{51}$$

using:

$$\mathbb{P}_1(\Sigma_B = l) = \binom{c-1}{l-1} p^{l-1}(1-p)^{c-l}, \tag{52}$$

and:

$$\mathbb{P}_0(\Sigma_B = l) = \binom{c-1}{l} p^{l}(1-p)^{c-l-1}. \tag{53}$$

Eq. (45), (50) and (51) are consequently used to compute the binary entropy functions of Eq. (16) which have to be averaged over f_P using numerical integration.

Acknowledgment

We would like to thank the reviewers and Teddy Furon from INRIA for their insightful and suggestive comments on this article.

References

[1] N. R. Wagner, "Fingerprinting," in *SP '83: Proceedings of the 1983 IEEE Symposium on Security and Privacy.* Washington, DC, USA: IEEE, 1983, p. 18.

[2] J. K. Milen, *Discussion*, ser. Foundations of Secure Computations. R. A. DeMillo et. al., Academic Press, 1978.

[3] G. Blakley, C. Meadows, and G. Purdy, "Fingerprinting long forgiving messages," in *of Crypto '85 Springer-Verlag Berlin, Heidelberg.* Springer, 1986, pp. 180–189. [Online]. Available: http://www.springerlink.com/index/H273724255J45015.pdf

[4] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *in Proc. Advances in Cryptology-Crypto '94: Springr-Verlag, 1994, LNCS*, vol. 839pp. Springer, 1994, pp. 257–270. [Online]. Available: http://www.springerlink.com/index/64Q2TG3PVDJ69X7J.pdf

[5] G. Tardos, "Optimal probabilistic fingerprint codes," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, vol. 55, no. 2. ACM, May 2003, pp. 116–125. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1346330.1346335

[6] 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*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003, pp. 472–479. [Online]. Available: http://portal.acm.org/citation.cfm?id=644187

[7] B. Škorić, T. U. Vladimirova, M. Celik, and J. C. Talstra, "Tardos Fingerprinting is Better Than We Thought," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3663–3676, Aug. 2008.

[8] T. Furon, L. Pérez-Freire, and A. Guyader, "Estimating the minimal length of Tardos code," *Information Hiding*, vol. 5806, pp. 176–190, 2009. [Online]. Available: http://www.springerlink.com/index/06125X51892M1743.pdf

[9] T. Furon, A. Guyader, and F. Cérou, "On the design and optimisation of Tardos probabilistic fingerprinting codes," in *Information Hiding*, ser. Lecture Notes in Computer Science, vol. 5284. Springer Berlin / Heidelberg, 2008, pp. 341–356.

[10] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of discrete Tardos fingerprinting codes," *Designs, Codes and Cryptography*, vol. 52, no. 3, pp. 339–362, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10623-009-9285-z

[11] T. Furon and L. Pérez-Freire, "Worst case attacks against binary probabilistic traitor tracing codes," in *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*. IEEE, 2009, pp. 56–60. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5386484 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5386484

[12] Y. W. Huang and P. Moulin, "Saddle-point solution of the fingerprinting capacity game under the marking assumption," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 2256–2260. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5205882

[13] M. Wu and Z. J. Wang, "Collusion resistance of multimedia fingerprinting using orthogonal modulation," in *IEEE Trans. Im-*

age Process. Citeseer, 2005, pp. 804–821. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.6433

[14] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, Apr. 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1188750

[15] F. Xie, T. Furon, and C. Fontaine, "On-off keying modulation and tardos fingerprinting," in *Proceedings of the 10th ACM workshop on Multimedia and security.* New York, New York, USA: ACM, 2008, pp. 101–106. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1411328.1411347 http://portal.acm.org/citation.cfm?id=1411328.1411347

[16] T. Kalker, "Considerations on watermarking security," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on.* IEEE, 2001, pp. 201–206. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=962734

[17] A. Kerckhoffs, "La Cryptographie militaire," *Journal des Sciences militaires*, vol. IX, pp. 5–38, 1883.

[18] P. Bas and A. Westfeld, "Two key estimation techniques for the broken arrows watermarking scheme," in *Proceedings of the 11th ACM workshop on Multimedia and security.* New York, NY, USA: ACM, 2009, pp. 1–8. [Online]. Available: http://portal.acm.org/citation.cfm?id=1597819

[19] B. Mathon, P. Bas, F. Cayre, and B. Macq, "Comparison of secure spread-spectrum modulations applied to still image watermarking," *Annals of Telecommunications - Annales Des Télécommunications*, Jul. 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/s12243-009-0119-9

[20] P. Bas and G. Doërr, "Practical security analysis of dirty paper trellis watermarking," in *Proceedings of the 9th international conference on Information hiding.* Springer-Verlag, 2007, pp. 174–188. [Online]. Available: http://portal.acm.org/citation.cfm?id=1782871

[21] B. Mathon, P. Bas, F. Cayre, and B. Macq, "Security and robustness constraints for spread-spectrum Tardos fingerprinting," in *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on.* Seattle, US: IEEE, Oct. 2010, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5711457

[22] B. Škorić, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Designs, Codes and*

Cryptography, vol. 46, no. 2, pp. 137–166, 2008. [Online]. Available: http://www.springerlink.com/index/241h612l378g3m72.pdf

[23] M. Kuribayashi and M. Morii, "On the systematic generation of Tardos's fingerprinting codes," *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pp. 748–753, Oct. 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4665174

[24] P. Moulin, "Universal fingerprinting: Capacity and random-coding exponents," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on.* IEEE, 2008, pp. 220–224. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4594980

[25] T. Furon, A. Guyader, and F. Cérou, "Decoding Fingerprinting Using the Markov Chain Monte Carlo Method," in *WIFS - IEEE Workshop on Information Forensics and Security.* Tenerife, Spain: IEEE, Dec. 2012. [Online]. Available: http://hal.inria.fr/hal-00757152

[26] M. Kuribayashi, "Bias equalizer for binary probabilistic fingerprinting codes," in *Proc. of 14th Information Hiding Conference, ser. LNCS, S. Verlag, Ed., Berkeley, CA, USA*, 2012.

[27] P. Bas and F. Cayre, "Achieving subspace or key security for woa using natural or circular watermarking," in *Proceedings of the 8th workshop on Multimedia and security.* ACM, 2006, pp. 80–88. [Online]. Available: http://portal.acm.org/citation.cfm?id=1161381

[28] K. Nuida, "Short collusion-secure fingerprint codes against three pirates," *International Journal of Information Security*, vol. 11, no. 2, pp. 85–102, Feb. 2012. [Online]. Available: http://link.springer.com/article/10.1007/s10207-012-0155-8

[29] D. Boesten and B. Škorić, "Asymptotic fingerprinting capacity in the Combined Digit Model," *Cryptology ePrint Archive: Report 2012/184*, 2012. [Online]. Available: https://eprint.iacr.org/2012/184.pdf

[30] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[31] M. Kuribayashi, "Interference Removal Operation for Spread Spectrum Fingerprinting Scheme," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 403–417, Apr. 2012. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6032747

[32] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *Computer Journal*, vol. 14, pp. 422–425, 1971.

[33] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, 1997. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=650120

[34] J. Cao and J. Huang, "Controllable Secure Watermarking Technique for Tradeoff Between Robustness and Security," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 821–826, Apr. 2012.

[35] M. Wu, W. Trappe, Z. J. Wang, and K. R. Liu, "Collusion-resistant fingerprinting for multimedia," *IEEE Signal Process. Mag.*, vol. 21, no. 2, pp. 15–27, 2004.

[36] L. Pérez-Freire, F. Pérez-González, and T. Furon, "On achievable security levels for lattice data hiding in the Known Message Attack scenario," in *Proceedings of the 8th workshop on Multimedia and security.* ACM, 2006, pp. 68–79.

[37] M. Kuribayashi, "Tardos's Fingerprinting Code over AWGN Channel," in *Information Hiding*, ser. Lecture Notes in Computer Science, R. Böhme, P. Fong, and R. Safavi-Naini, Eds. Springer Berlin Heidelberg, 2010, vol. 6387, pp. 103–117.