On the design and optimisation of Tardos probabilistic fingerprinting codes

Teddy Furon, Arnaud Guyader, Frédéric Cérou

On the design and optimization of Tardos probabilistic fingerprinting codes

Teddy Furon, Arnaud Guyader, and Frédéric Cérou

Centre de Recherche INRIA Rennes Bretagne Atlantique*

Abstract. G. Tardos [1] was the first to give a construction of a fingerprinting code whose length meets the lowest known bound in $O(c^2 \log \frac{n}{\epsilon_1})$. This was a real breakthrough because the construction is very simple. Its efficiency comes from its probabilistic nature. However, although G. Tardos almost gave no argument of his rationale, many parameters of his code are precisely fine-tuned. This paper proposes this missing rationale supporting the code construction. The key idea is to render the statistics of the scores as independent as possible from the collusion process. Tardos optimal parameters are rediscovered. This interpretation allows small improvements when some assumptions hold on the collusion process.

1 In Gabor Tardos' shoes

This article deals with active fingerprinting, also known as traitor tracing, or forensics, when applied on multimedia content. Fingerprinting is the application where a content server distributes personal copies of the same content to n different buyers. Some are dishonest users, called colluders, who mix their copies to yield a pirated content.

A binary fingerprinting code is a set of n different m bit sequences $\{\mathbf{X}_j\}_{j=1}^n$. Each sequence identifying a user has to be hidden in his/her personal copy with a watermarking technique. When a pirated copy is found, the server retrieves a m bit sequence $\mathbf{Y}$ and accuses some users or nobody. There are two kinds of errors: accusing an innocent (i.e. a false positive whose probability is denoted p_{fp}) and accusing none of the colluders (i.e. a false negative with probability p_{fn}). The designers of the fingerprinting code must assess the minimum length of the code so that the probabilities of error are below some significance levels: $p_{fa} < \epsilon_1$ and $p_{fn} < \epsilon_2$. One of the best fingerprinting codes is a probabilistic code proposed by G. Tardos, where $m = O(c^2 \log \frac{n}{\epsilon_1})$, where c is the number of colluders. Before Tardos' work, the existence of such a short code was proven. Tardos is the first to exhibit a construction which is surprisingly simple.

This breakthrough has appealed a numerous amount of research works. The goal of Tardos was to show a construction of an efficient fingerprinting code, where the length of the code was approximated. This yields a thread of works about refining the lower bound of the code length [2–4]. Improvements, such as

* This work is supported by the French national programme "Securité ET INformatique" under project NEBBIANO, ANR-06-SETIN-009.

symmetric codes and q-ary codes have been proposed in [2]. Indeed, we study in this paper Skoric's symmetric version of the binary Tardos code. In the sequel, we abusively shorten this expression in 'Tardos code' in reference to the inventor of this family of codes. Implementation issues have also been addressed in [4, 5].

1.1 A pedagogical approach

The nature of our article is very different. We are not providing better estimations of the code length. Our primary goal is to understand the key idea behind this code. Reading Tardos' seminal paper is very frustrating because Tardos proposes his construction hardly giving any clue on his rationale. His paper shows the code, gives a rough expression of its length, and chiefly proves that the probabilities of error match the constraints. Our aim is to rediscover how Tardos came up to invent this code. We have found different interpretations or explanations than those given in the previous works on Tardos codes, and we believe they help understanding how probabilistic codes work. Thanks to this better understanding, some small improvements are given at the end of the paper.

Another product of our analysis is that we create different classes of collusion. Usually, cryptographic papers dealing with traitor tracing use classes of collusion based on the nature of the symbols the collusion can forge: these are commonly denoted narrow or wide sense classes [6]. Nevertheless, these two classes are equivalent for binary codes studied in this article. Another variation is whether or not the collusion can produce erasures in detectable [7] or even undetectable positions [8]. We do not consider erasures here. We base the introduced four new classes of collusion not on the nature of the potentially forged symbols, but on side information the collusion has access to (see section 2 for details).

1.2 Probabilistic codes

We keep the same structure of code than G. Tardos' one, what he defined as probabilistic codes. We denote random variables by capital letters and their occurences by their normal version. The sequence $\mathbf{X_j} = (X_{j1}, \cdots, X_{jm})$ identifying user j is composed of m independent binary symbols, with $P(X_{ji} = 1) = p_i$, $\forall i \in [m]$, with $[m]$ denoting $\{1, \ldots, m\}$. $\{P_i\}_{i=1}^m$ are independent and identically distributed random variables in the range $[0, 1]$: $P_i \sim f(p)$. They have been drawn, taking the values $\{p_i\}_{i=1}^m$, before the code construction. $\mathbf{Y} = (Y_1, \cdots, Y_m)$ is the sequence decoded from the pirated copy. The accusation process accuses user j if $S_j > T$, where $S_j = \sum_{i=1}^m U(Y_i, X_{ji}, p_i)$, and T is a threshold. Roughly speaking, $U(Y_i, X_{ji}, p_i)$ is positive when $X_{ji} = Y_i$, tending to accuse user j, $U(Y_i, X_{ji}, p_i)$ is negative when $X_{ji} \neq Y_i$, tending to plead user j as innocent.

The random variables $\{P_i\}_{i=1}^m$ are independent and identically distributed with a pdf f which is assumed to be symmetric around 1/2: $f(p) = f(1-p)$. It means that symbols '1' and '0' play a similar role with probability p or $1-p$. This implies that the summands $U(Y_i, X_{ji}, p_i)$ in the accusation sum S_j follow

the rules:

$$U(1,1,p) = g_1(p), \quad U(0,0,p) = g_1(1-p), \tag{1}$$
$$U(1,0,p) = g_0(p), \quad U(0,1,p) = g_0(1-p). \tag{2}$$

Functions g_0 and g_1 are notations from G. Tardos' paper, and common to Skoric's works [2,3]. The function g_1 is used when symbols X_{ji} and Y_i match, function g_0 when they are different. These weights depend on the probability to find symbol Y_i at this index, i.e. p_i if $Y_i = 1$, $1 - p_i$ if $Y_i = 0$.

Designing a binary probabilistic code means to find the optimal functions f, g_1 and g_0. But, the definition of optimality is not very clear. Skoric *et al.* took Chernoff's lower bound of the code length as the criterion, and they partially rediscovered Tardos functions [3]. We invoke a different rationale.

The main drawback of probabilistic codes is that, a priori, the score of the innocents is a random variable whose statistics depend on the collusion process, which is unknown at the decoding side. This also obviously holds for the scores of the colluders. On the other hand, Tardos announces two very astonishing facts:

1. The probabilities of error are guaranteed whatever the collusion process is. The threshold and the length are also fixed and independent of the collusion process (for a given collusion size c).
2. There is no need to calculate the scores of all the users. A user is deemed guilty if his accusation sum is greater than the threshold.

Although Tardos did not say anything on the key ideas supporting his code construction, we believe that his intuition was to render the scores as independent as possible from the collusion process. In other words, the first task is to render the statistics of the scores independent, before optimizing the code length.

The fingerprinting code being probabilistic, it is not surprising that our study is solely based on the statistics of the score of an innocent (section 3) and of the score of a colluder (section 4). The score being a sum of independent random variables, we consider it as Gaussian distributed. This assumption really helps interpretating what G. Tardos had in mind. Item number one in the list above implies that we need expectations and variances of the scores (innocents and colluders) to be independent of the collusion process. Item number two implies that the scores are also mutually independent. Tardos' choices of functions f, g_1 and g_0 do fulfill these two conditions. This paper studies the converse: we look for functions f, g_1 and g_0 achieving independence.

However, this use of the central limit theorem is absolutely not recommended when estimating the code length because it amounts to integrate the distribution function on its tail where the Gaussianity assumption does not hold. In other words, when the expectations and variances are set independent of the collusion process, the behaviour of the code is fixed up to the second order. Nevertheless, a precise evaluation of the threshold value and the code length would need further developments. The Berry-Esséen bound shows that the gap between the

Gaussian law and the real distribution of the scores depends on their third moment, which a priori depends on the collusion process[1].

2 The marking assumption and the four classes of collusion

Fingerprinting code has been first studied by the cryptographic community. The marking assumption was a concept invented by Boneh and Shaw [7]. It states that, in its narrow-sense version, whatever the strategy of the collusion $\{j_1, \cdots, j_c\}$, we have $Y_i \in \{X_{j_1 i}, \cdots, X_{j_c i}\}$. In words, colluders forge the pirated copy by assembling chunks from their personal copies. It implies that if, at chunk i, the colluders symbols are identical, then this symbol value is decoded at the i-th chunk of the pirated copy.

This is what watermarkers have understood from the pioneering cryptographic work. However, this has led to misconceptions. Another important thing is the way cryptographers have modelized a host content: it is a binary string where some symbols can be changed without spoiling the regular use of the content. These locations are used to insert the sequence symbols. This implies that colluders disclose symbols from their identifying sequences comparing their personal copies symbol by symbol. Is this the case with multimedia fingerprinting?

In a multimedia application, the content is divided into chunks c_i. A chunk can be a few second clip of audio or video. Symbol X_{ji} is hidden in the i-th chunk of the content with a watermarking technique. This gives the i-th chunk sent to the j-th user: c_{ji}. In this paper, we only address the collusion process where the pirated copy is forged by picking chunks from the colluders personal copies. We do not address mixing of several chunks into one.

2.1 The blind colluders

We consider a first class of colluders. Before receiving the personal copies, these c dishonest users, denoted by their indices $\{j_1, \cdots, j_c\}$, have already agreed on how to forge the pirated copy. This strategy amounts to set an assignation sequence $(M_1, \cdots, M_m)$ with $M_i \in \{j_1, \cdots, j_c\}$, such that $Y_i = X_{M_i,i}$. We assume that the colluders share the risk, so that the cardinal $|\{i|M_i = j_u\}| \approx m/c$, for all $u \in [c]$. The assignation sequence is random and independent of the personal copies. We introduce the random variable Σ_i as the number of symbols '1' which the collusion gets at index i, the conditional probability concerning Y_i is given by: $P_{Y_i}(0|\Sigma_i = \sigma_i) = (c - \sigma_i)/c$. The important thing is that the blind colluders set their assignation sequence without observing their personal copies.

2.2 The sighted colluders

This second class of colluders differs in the fact that the assignation sequence is now a function of the personal copies. These colluders are able to split their copies

[1] Skoric *et al.* also stress the crucial role of the cutoff parameter in the convergence speed of the Berry-Esséen bound [2, Sec. 7.1].

in chunks and to compare them sample by sample. Hence, for any index i, they are able to notice that, for instance, chunks $c_{j_1 i}$ and $c_{j_2 i}$ are different or identical. For binary embedded symbols, they can constitute two stacks, each containing identical chunks. This allows new collusion processes such as a majority vote (the pirated chunk is taken for the stack whose size is bigger) or a minority vote... From a statistical point of view, the majority vote yields the following conditional probability: $P_{Y_i}(0|\Sigma_i = \sigma_i) = 1$ if $\sigma_i < c/2$, 0 else. For the minority vote: $P_{Y_i}(0|\Sigma_i = \sigma_i) = 1$ if $\sigma_i > c/2$, 0 else, with the noticeable exceptions due to the marking assumption: $P_{Y_i}(0|\Sigma_i = 0) = 1$ and $P_{Y_i}(0|\Sigma_i = c) = 0$.

The important thing is that colluders can notice differences between chunks, but they cannot tell which chunk contains symbol '0'. Hence, symbols '1' and '0' play a symmetric role, which strongly links the conditional probabilities:

$$P_{Y_i}(0|\Sigma_i = \sigma_i) = P_{Y_i}(1|\Sigma_i = c - \sigma_i) = 1 - P_{Y_i}(0|\Sigma_i = c - \sigma_i). \quad (3)$$

2.3 The cryptographic colluders

In the third class, the colluders know parts of their code sequences. This is the case in the model used by cryptographers since Boneh and Shaw [7]. The bits are directly pasted in the host content string, and thus observable by the colluders. However, the marking assumption is still valid. New strategies are then possible like the 'All 0' (resp. 'All 1') consisting in putting a symbol '0' (resp. '1') in the pirated copy chunks whenever this is possible. This is the case when all the colluders do not have c embedded '1' (resp. '0') in their chunks: $P_{Y_i}(0|\Sigma_i = \sigma_i) = 1$ if $\sigma_i < c$, 0 else (resp. $P_{Y_i}(0|\Sigma_i = \sigma_i) = 0$ if $\sigma_i > 0$). Note that the relationship (3) does not hold anymore.

2.4 The omniscient colluders

In this last class, the colluders know the value of p_i for all index i. They can adapt their strategy for each index chunk according to its value p_i. From a statistical point of view, we just write that the conditional probabilities depend on σ_i and p_i: $P_{Y_i}(0|\Sigma_i = \sigma_i, P_i = p_i)$. Subsection 4.2 reveals the optimum values of the worst case collusion. This class of collusion breaks the code, in the sense that the omniscient colluders will not be accused almost surely. Therefore, it is mandatory that the value of p_i for all index i remains secret, so that, this class of colluders never exists. This threat considerably reduces the interest in making P_i discrete random variables as proposed in [4]. This option shortens the size of memory needed to store the value of p_i, but it introduces a security flaw as the set of possible values is public.

2.5 An open issue about multimedia fingerprint

Cryptographic fingerprinting codes and probabilistic codes target the third type of collusion. We do think that such powerful colluders are not realistic in multimedia fingerprinting. Colluders do not have the watermark decoder to disclose

their embedded sequence. An open question is then: Provided the colluders are less powerful than foreseen, is there a hope to invent more suitable (i.e. shorter sequences) fingerprinting code?

3 User j is an innocent

In our quest of finding the optimal functions g_1, g_0 and f, we apply the idea of setting the statistics of S_j (knowing that user j is innocent) independent from the collusion process. We assume that symbols $\{Y_i\}_{i=1}^m$ are mutually independent and distributed such that $P(Y_i = 1) = q_i$ (denoted Q_i, when considered as a random variable). This distribution a priori depends on several parameters: the size of the collusion, the class of the collusion, the collusion process used at index i and the value of p_i.

3.1 Expectation of S_j

$\mathbb{E}(S_j) = \sum_{i=1}^m \mathbb{E}(U(Y_i, X_{ji}, P_i)) = m\mathbb{E}(U(Y_i, X_{ji}, P_i))$, where $\mathbb{E}$ denotes mathematical expectation. Dropping the subscripts and using (1) and (2):

$$\begin{aligned}\mathbb{E}(U(Y,X,P)) &= \mathbb{E}_P\mathbb{E}_Y\mathbb{E}_X U(Y,X,P)\\ &= \int_0^1 q(pg_1(p) + (1-p)g_0(p))f(p)dp\\ &+ \int_0^1 (1-q)(pg_0(1-p) + (1-p)g_1(1-p))f(p)dp. \qquad (4)\end{aligned}$$

Now, the problem is that the detection side ignores many parameters of the distribution, so that q is unknown. We believe that Tardos had in mind to remove this dependence on q. The most general manner to achieve this, is the following:

$$\mathbb{E}(U(Y,X,P)) = \mathbb{E}(H(P)) + \mathbb{E}(QA(P)),$$

with

$$H(p) = pg_0(1-p) + (1-p)g_1(1-p), \qquad (5)$$
$$A(p) = H(1-p) - H(p). \qquad (6)$$

The expectation of the summand of the innocent's score is independent of the collusion process if and only if $A(p) = 0$, $\forall p \in [0,1]$. This implies that H is symmetric: $H(p) = H(1-p)$, $\forall p \in [0,1]$. In this case:

$$\mathbb{E}(U(Y,X,P)) = \int_0^1 H(p)f(p)dp.$$

Hence, we know in advance the expectation of the innocent's score, regardless of the collusion process. If it is not zero, we can always subtract this constant to the scores. Hence, without loss of generality, let us impose that $\mathbb{E}H(P) = 0$.

3.2 Variance of S_j

The variance of a random variable a is denoted by $\mathrm{Var}(a)$. The summands of the score are independent and their expectation is null. Hence: $\mathrm{Var}(S_j) = \mathbb{E}(S_j^2) = \sum_{i=1}^{m} \mathbb{E}(U(Y_i, X_{ji}, P_i)^2)$.

$$\begin{aligned}\mathbb{E}(U(Y,X,P)^2) &= \mathbb{E}_p\mathbb{E}_Y\mathbb{E}_X U(Y,X,P)^2 \\ &= \int_0^1 q(pg_1(p)^2 + (1-p)g_0(p)^2)f(p)dp \\ &+ \int_0^1 (1-q)(pg_0(1-p)^2 + (1-p)g_1(1-p)^2)f(p)dp. \quad (7)\end{aligned}$$

The most general manner to achieve independence is as follows:

$$\mathbb{E}(U(Y,X,P)^2) = \mathbb{E}(G(P)) + \mathbb{E}(QB(P)),$$

with

$$G(p) = pg_0(1-p)^2 + (1-p)g_1(1-p)^2, \quad (8)$$
$$B(p) = G(1-p) - G(p). \quad (9)$$

The variance of the summand of the innoncent's score is independent of the collusion process if and only if $B(p) = 0, \forall p \in [0,1]$. This implies that G is symmetric: $G(p) = G(1-p)$, $\forall p \in [0,1]$. In this case:

$$\mathbb{E}(U(Y,X,P)^2) = \int_0^1 G(p)f(p)dp.$$

Hence, we know in advance the variance of the innocent's score, regardless of the collusion process. If it is not one, we can always normalize the scores. Hence, without loss of generality, let us impose that $\mathbb{E}G(P) = 1$ so that $\mathrm{Var}(S_j) = m$.

3.3 Cross-correlation

Thanks to the CLT, when m is big enough, the score S_j for an innocent is approximately distributed as a Gaussian distribution $\mathcal{N}(0, m)$. We investigate here the dependence with the score S_k knowing that user k is also an innocent. This amounts to calculate their correlation since S_k and S_j are deemed Gaussian.

$$\mathrm{Cov}(S_j, S_k) = \mathbb{E}(S_j S_k) = \sum_{i=1}^{m} \mathbb{E}(U(Y_i, X_{ji}, P_i)U(Y_i, X_{ki}, P_i)), \text{ with}$$

$$\begin{aligned}\mathbb{E}(U(Y,X_j,P)U(Y,X_k,P)) &= \mathbb{E}(Q(Pg_1(P) + (1-P)g_0(P))^2) \\ &+ \mathbb{E}((1-Q)(Pg_0(1-P) + (1-P)g_1(1-P))^2) \\ &= \mathbb{E}(H^2(P) + Q(H^2(1-P) - H^2(P))). \quad (10)\end{aligned}$$

A cross correlation independent of the collusion process is achieved for $H^2(p) = H^2(1-p)$. This condition was already fulfilled, *cf.* subsection 3.1. Thus we have $\mathbb{E}(U(Y, X_j, P)U(Y, X_k, P)) = \mathbb{E}(H^2(P))$, and scores S_j and S_k are independent if and only if $H(p) = 0$, $\forall p \in [0, 1]$. Injecting this new result in the expression of $G(p)$ gives:

$$G(p) = p.\frac{(1-p)^2}{p^2} g_1^2(1-p) + (1-p)g_1^2(1-p) = \frac{1-p}{p} g_1^2(1-p). \tag{11}$$

Remembering the constraint on the variance of the innocent score, we have:

$$\int_0^1 g_1^2(1-p)\frac{1-p}{p} f(p)dp = \int_0^1 g_1^2(p)\frac{p}{1-p} f(p)dp = 1. \tag{12}$$

We have also collected two other equations, $\forall p \in [0, 1]$:

$$H(p) = 0 \;\rightarrow\; pg_0(1-p) = -(1-p)g_1(1-p) \tag{13}$$

$$G(p) = G(1-p) \;\rightarrow\; (1-p)g_1(1-p) = pg_1(p) \tag{14}$$

In this last equation, we assume that g_1 as a constant sign over $[0, 1]$. We do not have enough equations to fully determine the three functions g_1, g_0, and f. However, any occurence of $g_0(p)$, $g_0(1-p)$, and $g_1(1-p)$ can be replaced by expressions involving only $g_1(p)$.

4 User j is a colluder

We seek more relations to find out the three functions. For the moment, we do not restrict our study to a particular collusion process.

4.1 Variance of S_j

The collusion process implies a distribution of the couple $\{Y_i, X_{ji}\}$, with user j being a colluder. Dropping the subscript but keeping in mind the dependence on p, this distribution equals $P_{YX}(y, x|p) = P_Y(y|x, p)P_X(x|p)$. Colluders do not receive sequences different in nature than the ones of the innocents: $P_X(x|p) = p^x(1-p)^{1-x}$. Finally, we can write:

$$\begin{aligned} P_{YX}(0,0|p) &= P_Y(0|0,p)(1-p) & P_{YX}(1,1|p) &= P_Y(1|1,p)p \\ P_{YX}(1,0|p) &= (1-P_Y(0|0,p))(1-p) & P_{YX}(0,1|p) &= (1-P_Y(1|1,p))p \end{aligned}$$

Hence, the collusion process is only defined, from a statistical point of view, via two functions depending on p: $P_Y(0|0, p)$ and $P_Y(1|1, p)$. As done for the variance of the score of an innocent, we write:

$$\begin{aligned} \mathbb{E}(U(Y, X, P)^2) = \mathbb{E}(&P_{YX}(0,0|P)g_1^2(1-P) + P_{YX}(1,1|P)g_1^2(P)) \\ &+ \mathbb{E}(P_{YX}(0,1|P)g_0^2(1-P) + P_{YX}(1,0|P)g_0^2(P)). \end{aligned}$$

Knowing relations (13) and (14), we express the four summands with $g_1(p)$:

$$\begin{aligned}
P_{YX}(0,0|p)g_1^2(1-p) &= P_Y(0|0,p)(1-p)g_1^2(1-p) = P_Y(0|0,p)\frac{p^2}{(1-p)}g_1^2(p),\\
P_{YX}(1,1|p)g_1^2(p) &= P_Y(1|1,p)pg_1^2(p),\\
P_{YX}(0,1|p)g_0^2(1-p) &= P_Y(0|1,p)pg_1^2(p) = (1-P_Y(1|1,p))pg_1^2(p),\\
P_{YX}(1,0|p)g_0^2(p) &= (1-P_Y(0|0,p))(1-p)g_0^2(p)\\
&= (1-P_Y(0|0,p))\frac{p^2}{1-p}g_1^2(p).
\end{aligned}$$

It appears that, whatever the collusion process, the expectation of the square of the summands in the colluders' score is constant:

$$\mathbb{E}(U(Y,X,P)^2) = \mathbb{E}\left(g_1(P)^2\frac{P}{1-P}\right) = 1$$

Thus, given the results of Sect. 3, the variance of the score is already independent of the collusion process, equaling $\mathrm{Var}(S_j|j\in\mathcal{C}) = m(1-\mathbb{E}(U(Y,X,P))^2)$. Hence, the impact of the collusion process is solely determined by $\mathbb{E}(U(Y,X,P))$: The lower this expectation is, the more difficult it will be to find back the colluders.

4.2 Expectation of S_j

The expectation of the score of one colluder is surprisingly much more involved. However, as G. Tardos did, it is simpler to calculate the expectation of the sum of the c scores of the colluders. If the colluders share the risk evenly, we can suppose that the expectation of one colluder's score is the average of this expectation: $\mathbb{E}(S_j|j\in\mathcal{C}) = c^{-1}\mathbb{E}(\sum_{k=1}^{c}U(Y,X_{j_k},P))$. If this is not the case, we are sure that at least one colluder has an expected score bigger than this average. Hence, at the decoding side, it will be easier to distinguish the two hypothesis (j is an innocent, j is a colluder) when this expectation is bigger. We have:

$$\begin{aligned}
\mathbb{E}\left(\sum_{k=1}^{c}U(Y,X_{j_k},P)\right) = \mathbb{E}\Bigg(&\sum_{\sigma=0}^{c}P_{Y\Sigma}(0,\sigma|P)((c-\sigma)g_1(1-P)+\sigma g_0(1-P))\\
&+\sum_{\sigma=0}^{c}P_{Y\Sigma}(1,\sigma|P)((c-\sigma)g_0(P)+\sigma g_1(P))\Bigg).
\end{aligned} \tag{15}$$

As usual, we express the summands with function $g_1(P)$.

$$\mathbb{E}\left(\sum_{k=1}^{c}U(Y,X_{j_k},P)\right) = \mathbb{E}\frac{g_1(P)}{1-P}\sum_{\sigma=0}^{c}(P_Y(0|\sigma,P)-P_Y(1|\sigma,P))P_\Sigma(\sigma)(cP-\sigma), \tag{16}$$

with $P_\Sigma(\sigma) = \binom{c}{\sigma}P^\sigma(1-P)^{c-\sigma}$.

The omniscient colluders This last equation shows what the worst collusion process is. The goal of the colluders is to minimize this expectation, which happens if $P_Y(0|\sigma,p) = 1$ and $P_Y(1|\sigma,p) = 0$ when $cp < \sigma$, $P_Y(0|\sigma,p) = 0$ and $P_Y(1|\sigma,p) = 1$ else. This optimum strategy is only possible when the collusion knows exactly the values σ_i (number of symbols '1' it got) and $\{p_i\}_{i=1}^m$. This corresponds to the omniscient colluders class. The next subsection deals with the other classes of colluders.

4.3 Independence from the collusion strategy

The other classes of colluders have in common their ignorance of the values $\{p_i\}_{i=1}^m$. This translates in the fact that P can be forgotten in the conditional probabilities. Hence, we can exchange the expectation and sum in (16):

$$\mathbb{E}\left(\sum_{k=1}^{c} U(Y, X_{j_k}, P)\right) = \sum_{\sigma=0}^{c} (P_Y(0|\sigma) - P_Y(1|\sigma))\binom{c}{\sigma} I_c(\sigma) \tag{17}$$

with

$$\begin{aligned} I_c(\sigma) &= \mathbb{E}\left(g_1(P)P^\sigma(1-P)^{c-\sigma-1}(cP-\sigma)\right), \\ &= \int_0^1 \frac{g_1(p)f(p)}{(1-p)} p^\sigma (1-p)^{c-\sigma}(cp-\sigma)dp \end{aligned} \tag{18}$$

This family of integrals has the following property:

Lemma 1. *c and σ being integers such that $0 \le \sigma \le c$, we have:*

$$I_c(\sigma) = -I_c(c-\sigma),$$

Therefore, $\sum_{\sigma=0}^{c}\binom{c}{\sigma} I_c(\sigma) = 0$ and $I_c(c/2) = 0$ when c is even.

The proof of this lemma is based on the change of variable $p' = 1-p$ in the integral, knowing that this change lets $g_1(p)f(p)(1-p)^{-1}$ invariant according to (14) and the assumption that f is symmetric around $1/2$.

Inserting $P_Y(1|\sigma) = 1 - P_Y(0|\sigma)$ in (17), simplifies in:

$$\mathbb{E}\left(\sum_{k=1}^{c} U(Y, X_{j_k}, P)\right) = 2\sum_{\sigma=0}^{c} P_Y(0|\sigma)\binom{c}{\sigma} I_c(\sigma). \tag{19}$$

The decoding side a priori ignores the values $\{P_Y(0|\sigma)\}_{\sigma=0}^c$, except for two cases: The marking assumption states that $P_Y(0|\sigma=0) = 1$, and $P_Y(0|\sigma=c) = 0$. Hence, the only way to get rid of the remaining unknown conditional probabilities is to find function $g_1(p)f(p)$ such that $I_c(\sigma) = 0$, $\forall\sigma$, $0 < \sigma < c$. (14) tells us that the block $g_1(p)f(p)/(1-p)$ is symmetric. For this reason, we define a symmetric function $T(p) = pg_1(p)f(p)$ which could cancel the $(c-1)$ integrals: $\forall\sigma, 0 < \sigma < c$

$$I_c(\sigma) = \int_0^1 T(p)P_{c,\sigma}(p)dp \quad \text{with} \quad P_{c,\sigma}(p) = p^{\sigma-1}(1-p)^{c-\sigma-1}(cp-\sigma).$$

Lemma 2. *The family of polynomials* $\{P_{c,\sigma}\}_{\sigma=1}^{c-1}$ *spans the subspace* $\mathcal{P}$ *of polynomials of degree equal or less than* $(c-1)$ *whose integral over* $[0,1]$ *is null.*

The proof is in Annex A.1. A corollary of this proposition is that such a function T exists: $T(p) = cst$, $\forall p \in [0,1]$. The next subsection shows that Tardos actually made this choice while the last section of the paper considers other solutions.

4.4 Rediscovering Tardos' solution

Suppose we have found a function S, such that its projection onto $\mathcal{P}$ is null (ie. $I_c(\sigma) = 0$, $\forall 0 < \sigma < c$), it is symmetric and positive (ie. $S(p) = S(1-p) \geq 0$). This means that S is a good candidate as a prototype for $pg_1(p)f(p)$. Note that αS with $\alpha > 0$ also fulfills these requirements. What is the maximum α? Let us write $pf(p)g_1(p) = \alpha S(p)$. (12) and the fact that f is a pdf defines these two constraints for α:

$$\alpha^{-1} = \int_0^1 S(p)g_1(p)(1-p)^{-1}dp = \int_0^1 S(p)(pg_1(p))^{-1}dp$$

Thanks to the properties of S, the expectation of the sum of the colluders' score is now independent of their strategy and equal to:

$$\mathbb{E}\left(\sum_{k=1}^{c} U(Y, X_{j_k}, P)\right) = 2c\alpha \int_0^1 S(p)p^{c-1}dp, \tag{20}$$

$$= 2\frac{\int_0^1 S(p)dp}{\sqrt{\int_0^1 \frac{S(p)g_1(p)}{1-p}dp . \int_0^1 \frac{S(p)}{pg_1(p)}dp}} \tag{21}$$

The simplification of the numerator happens thanks to lemma 2. The polynomial $p^{c-1} - c^{-1}$ has a degree equal to $c-1$ and a null integral. Hence, it belongs to the linear subspace $\mathcal{P}$. S being orthogonal to this subspace (with a scalar product defined as the integral over $[0,1]$), we have $\int_0^1 S(p)p^{c-1}dp = c^{-1}\int_0^1 S(p)dp$.

At last, once we have rendered the statistics of the scores of the innocents and of the colluders independent from the collusion process, comes the criterion of optimality. We will define it as the choice of functions f and g_1 maximizing the expectation of the sum of the colluders. A Cauchy-Schwarz inequality on the product of the denominator shows the maximum:

$$\mathbb{E}\left(\sum_{k=1}^{c} U(Y, X_{j_k}, p)\right) \leq \frac{2\int_0^1 S(p)dp}{\int_0^1 S(p)(p(1-p))^{-1/2}dp} \tag{22}$$

with equality if $g_1(p) \propto \sqrt{(1-p)/p}$, and thus $f(p) \propto S(p)/\sqrt{p(1-p)}$. Note that this maximization holds for any choice of function S. For the simplest choice ($S(p) = 1$), we finally rediscover Tardos' functions:

$$f(p) = \frac{1}{\pi\sqrt{p(1-p)}}, \quad g_1(p) = \sqrt{\frac{1-p}{p}}.$$

This choice yields $\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, p)) = 2/\pi$.

5 How to make a better choice?

We investigate whether there is still room for improvements compared to Tardos choice.

5.1 Fixed size of collusion

If the application scenario can assess that the colluders will always be no more than c, then the dimension of the space $\mathcal{P}$ is finite and equal to $(c-1)$. In order to use powerful mathematical objects such as Legendre polynomials, a change of variable $p = (1+x)/2$ is advised. This transforms the rhs of (22) into:

$$\frac{\int_{-1}^{1} t(x)dx}{\int_{-1}^{1} t(x)(1-x^2)^{-1/2}dx}, \quad \text{with } t(x) = S((1+x)/2). \tag{23}$$

We are looking for the function t defined over $[-1,1]$ which maximizes this ratio, while having the following properties:

- $t(x) = t(-x)$ because $S(p)$ must be symmetric wrt $1/2$,
- its projection onto $\mathcal{P}$ is null because $I_c(\sigma) = 0$, $\forall 0 < \sigma < c$.
- $t(x) \geq 0$, $\forall x \in [-1,1]$,

Thanks to the change of variable, a basis of the linear space $\mathcal{P}$ is defined as polynomials of degree less or equal to $(c-1)$ whose integral over $[-1,1]$ are null, and orthonormal for the following scalar product: $< f, g >= \int_{-1}^{1} f(x)g(x)dx$. This exactly corresponds to the normalized Legendre polynomials[2]: $\{P_l^{\mathcal{L}}\}_{l=1}^{c-1}$. Odd order Legendre polynomials are anti-symmetric. Even order ones reach their maximum values at $x = \pm 1$, and $P_{2k}^{\mathcal{L}}(1) = 1$.

If t is smooth enough[3], it can be expressed as a series of Legendre polynomial: $t(x) = \sum_{k=0}^{\infty} \tau_k P_k^{\mathcal{L}}(x)$, $\forall x \in [-1,1]$. The above listed properties imply that: $\tau_{2k+1} = 0$ $\forall k$, $\tau_k = 0$ for $0 < k < c$, and $\tau_0 + \sum_{k \geq \lceil (c-1)/2 \rceil} \tau_{2k} \geq 0$.

Lemma 3. $\beta_l = \int_{-1}^{1} P_{2l}^{\mathcal{L}}(x)(1-x^2)^{-1/2}dx = \pi 2^{-4l}\binom{2l}{l}^2$.
Moreover: $0 < \beta_{l+1} < \beta_l$, $\forall l \in \mathbb{N}$.

Therefore:

$$\frac{\int_{-1}^{1} t(x)dx}{\int_{-1}^{1} t(x)(1-x^2)^{-1/2}dx} = \frac{2}{\pi}\left(1 + \tau_0^{-1} \sum_{k>(c-1)/2}^{\infty} \tau_{2k} 2^{-4k}\binom{2k}{k}^2\right)^{-1}, \tag{24}$$

which is maximized under the constraint $\tau_0 + \sum_{k>(c-1)/2} \tau_{2k} \geq 0$, by setting $\tau_{2k} = 0$, except $\tau_{2\lceil (c-1)/2 \rceil} = -\tau_0$. Table 1 gives the numerical values for the

[2] without the change of variable, we would have resorted to the shifted Legendre polynomials which are less known.

[3] This assumption prevents us from looking for discrete random variable $\{p_i\}$, *i.e.* when f is a sum of Dirac distributions as in [4].

first sizes of collusion. Note that these expectations converges to Tardos' one as c increases. In fact, when we cannot make any assumption on the maximum size of the collusions, Tardos' alternative $T(p) = pf(p)g_1(p) = 1$ is the only choice.

Another important fact is that the function S has an impact on f, but not on g_1 (see (22) and the line after). Hence, the assumptions yielding function S must be known when generating the code, but not necessary on the decoding side.

c	2	4	6	8	10
$m\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, p))$	85	74	71	69	68
Increase in %	33.3	16.4	10.8	8.1	6.5

Table 1. Best expectations of the sum of the scores of the colluders for a code length $m = 100$, when the size of the collusion is known. Corresponding increase in percentage compared to Tardos solution for which $m\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, P)) = 2m/\pi = 63.7$.

5.2 Unique collusion process

Suppose that we know what the collusion process is, *i.e.* we know all the probabilities $P_Y(1|\sigma, p)$ and $P_Y(0|\sigma, p)$ for any value of $p \in [0, 1]$ and $\sigma \in [c]$. We can simplify (16) in $\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, P)) = \mathbb{E}(g_1(P)(1-P)^{-1}C(P))$, where C, defined accordingly, solely depends on the collusion process. We can always decompose $C(p)$ as the sum of a symmetric function $C_s(p)$ and an anti-symmetric function $C_a(p)$. As we have constrained $g_1(p)(1-p)^{-1}$ to be symmetric, then $\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, P)) = \mathbb{E}(g_1(P)(1-P)^{-1}C_s(P))$. A Cauchy-Schwarz inequality gives an upper bound of this expectation, which holds for any function g_1:

$$\mathbb{E}\left(\frac{g_1(P)}{1-P}C_s(P)\right) \leq \sqrt{\mathbb{E}\left(\frac{P}{1-P}g_1^2(P)\right)}\sqrt{\mathbb{E}\left(\frac{C_s^2(P)}{(1-P)P}\right)} = \sqrt{\mathbb{E}\left(\frac{C_s^2(P)}{(1-P)P}\right)}.$$

The first square root equals one thanks to (14). Equality holds when $g_1(p) = \lambda C_s(p)p^{-1}$. The value of λ is given by the constraint on the variance of the score of the innocent: $\lambda = 1/\sqrt{\mathbb{E}(C_s^2(P)/P(1-P))}$, and the expectation is then equal to $\mathbb{E}(\sum_{k=1}^{c} U(Y, X_{j_k}, P)) = \lambda^{-1}$. In comparison, Tardos' choice yields a lower expectation:

$$\mathbb{E}\left(\sum_{k=1}^{c} U(Y, X_{j_k}, P)\right) = \mathbb{E}\left(\sqrt{\frac{C_s^2(P)}{P(1-P)}}\right) \leq \sqrt{\mathbb{E}\left(\frac{C_s^2(P)}{P(1-P)}\right)}. \tag{25}$$

Thanks to the concavity of the function $x \mapsto \sqrt{x}$, Jensen inequality ensures that our choice is better than Tardos' one with respect to the expectation of the sum of the scores of the colluders. Yet, this holds when the decoding side

exactly knows what function C is (collusion process and size of the collusion). It is a kind of matched detection process, such as the matched filters receptors in digital communications. This condition is very restrictive. However, there are some interesting points:

- Lemma 1 tells us that $I_2(1) = 0$, hence the value of $P_Y(0|\sigma = 1)$ has no impact when $c = 2$. The marking assumption fixes the remaining term $P_Y(0|\sigma = 0) = 1 - P_Y(0|\sigma = c) = 1$. Therefore, for a given function g_1, any collusion process involving only 2 colluders yields the same expectation $\mathbb{E}(\sum_{k=1}^{2} U(Y, X_{j_k}, P))$.
- When the collusion belongs to the blind class, the size of the collusion doesn't matter: $C(p) = C_s(p) = 2p(1-p)$, $\forall c \in \mathbb{N}$.

For these two conditions, the above maximization shows that $g_1(p) = \lambda(1-p)$. Table 5.2 shows the values of the expectation times the length of the code $m = 100$, for some functions C (column) against some functions g_1 optimal for a given collusion process (line). The last line corresponds to Tardos' choice. The scores on this line are all equal reflecting the independence versus the collusion process. The first line corresponds to the choice $g_1(p) = \lambda(1-p)$. It is extremely important to notice that expectations have been measured with $f(p) = (\pi\sqrt{p(1-p)})^{-1}$. In other words, there is still a degree of freedom to improve these scores. Or, we can say that the assumptions yielding the function C have an impact on g_1 but not f, such that they are needed on the decoding side but not on the coding side while generating matrix $\mathbf{X}$.

The scores on the diagonal are always the best score of a column as they correspond to matched accusation function and collusion process. However these functions g_1 are very sensitive with respect to the collusion: for instance, function g_1 tuned to fight against minority vote has excellent expectations when matched, but very bad scores when the collusion process is indeed a majority vote. The worst case attack led by omniscient colluders always has a dramatic effect. This stresses the fact that $\{p_i\}_{i=1}^{m}$ must absolutely remain secret.

6 Conclusion

The key idea supporting the probabilistic fingerprinting code proposed by G. Tardos is to render the statistics of the scores of the innocents and of the colluders independent from the collusion process. Achieving the independence for the first (expectation) and the second (variance) moments freezes all the degrees of freedom, determining the functions involved in the code. Tardos' choice is the most general. There is no room for improvements, except if the maximum collusion size is known when generating the code or if the collusion process is known at the decoding side.

	C																			
	class	blind	sighted						crypto						omniscient					
	process		Majority			Minority			All-1			All-0			Worst				
	size	$\forall c$	3	4	5	3	4	5	3	4	5	3	4	5	2	3	4	5	
g_1	blind	**71**	80	80	83	53	44	33	66	62	58	66	62	58	41	9	-19	-44	
	Maj-3	67	**84**	84	92	34	17	-5	59	50	43	59	50	43	44	9	-23	-51	
	Maj-4	67	84	**84**	92	34	17	-5	59	50	43	59	50	43	44	9	-23	-51	
	Maj-5	63	83	83	**93**	24	4	-23	53	43	35	53	43	35	45	9	-25	-54	
	Min-3	50	38	38	29	**75**	87	105	56	62	67	56	62	67	16	5	-2	-10	
	Min-4	43	27	27	18	74	**89**	111	51	58	65	43	58	65	11	3	0	-3	
	Min-5	40	24	24	15	73	89	**112**	48	57	63	48	57	63	9	2	0	-2	
	All1-3	69	73	73	73	62	59	55	**68**	66	64	68	66	64	36	8	-16	-37	
	All1-4	65	63	63	60	70	72	76	66	**67**	68	66	67	68	30	7	-12	-28	
	All1-5	59	53	53	47	73	80	90	63	66	**69**	63	66	69	25	6	-8	-21	
	All0-3	69	73	73	73	62	59	55	68	66	64	**68**	66	64	36	8	-16	-37	
	All0-4	65	63	63	60	70	72	76	66	67	68	66	**67**	68	30	7	-12	-28	
	All0-5	59	53	53	47	73	80	90	63	66	69	63	66	**69**	25	6	-8	-21	
	Tardos	64	64	64	64	64	64	64	64	64	64	64	64	64	32	7	-14	-32	

Table 2. Expectations of the sum of the scores of the colluders for a given collusion process C, a given size c and a matched accusation function g_1. $m = 100$. Expectations are in boldface font when the accusation function matches the collusion process.

A Proof of Lemmas

A.1 Lemma 2

We show that the family of polynomials $\{P_{c,\sigma}\}_{\sigma=1}^{c-1}$ spans the subspace $\mathcal{P}$ of polynomials of degree equal or less than $(c-1)$ whose integral over $[0,1]$ is null.

$\deg(P_{c,\sigma}) = c-1$ because $P_{c,\sigma}(p) = p^{\sigma-1}(1-p)^{c-\sigma-1}(cp-\sigma) = -\sigma p^{\sigma-1} + \ldots + (-1)^{c-\sigma-1}p^{c-1}$. Besides: $\int_0^1 P_{c,\sigma}(p)dp = c\frac{\sigma!(c-\sigma-1)!}{c!} - \sigma\frac{(\sigma-1)!(c-\sigma-1)!}{(c-1)!} = 0$.

Denote $N(p) = \sum_{\sigma=1}^{c-1}\alpha_\sigma P_{c,\sigma}(p) = \sum_{\ell=0}^{c-1}\beta_\ell p^\ell$ the null polynomial: $N(p) = 0$, $\forall 0 \leq p \leq 1$. The term in p^0 comes from the contribution of $\alpha_1 P_{c,1}(p)$. Hence $\beta_0 = 0$ implies $\alpha_1 = 0$. The term in p^1 comes from the contribution of $\alpha_1 P_{c,1}(p) + \alpha_2 P_{c,2}(p)$. Hence, $\beta_1 = 0$ implies $\alpha_2 = 0$ since $\alpha_1 = 0$, *etc.* Therefore, $\sum_{\sigma=1}^{c-1}\alpha_\sigma P_{c,\sigma} = 0$ implies $(\alpha_1, \ldots, \alpha_{c-1}) = \mathbf{0}$. This proves that the polynomials $\{P_{c,\sigma}\}_{\sigma=1}^{c-1}$ are linearly independent.

A.2 Lemma 3

We show that $\beta_l = \int_{-1}^{1} P_{2l}^{\mathcal{L}}(x)(1-x^2)^{-1/2}dx = \pi 2^{-4l}\binom{2l}{l}^2$, and $0 < \beta_{l+1} < \beta_l$.

Even degree Legendre polynomials have the following generic expression:

$$P_{2l}^{\mathcal{L}}(x) = 2^{-2l}\sum_{k=0}^{l}(-1)^{l-k}\frac{(2l+2k)!}{(l-k)!(l+k)!(2k)!}x^{2k} \quad . \tag{26}$$

Besides:

$$\int_{-1}^{1} \frac{x^{2k}}{\sqrt{1-x^2}} dx = \frac{\pi}{k!} \frac{1.3.5\ldots(2k-1)}{2^k} \quad . \tag{27}$$

Hence,

$$\beta_l = \frac{(-1)^l \pi}{2^{2l}} \sum_{k=0}^{l} c_{k,l} \quad \text{with } c_{k,l} = (-1)^k \frac{(2l+2k)!}{(l+k)!(l-k)!k!k!2^{2k}} \quad . \tag{28}$$

Note that $c_{0,l} = \binom{2l}{l}$, and $c_{k+1,l}/c_{k,l} = (k-l)(k+l+1/2)/(k+1)^2$. This means that β_l can be expressed thanks to an hypergeometric function of the second kind:

$$\beta_l = \frac{(-1)^l \pi}{2^{2l}} \binom{2l}{l} \, {}_2F_1(-l, l+1/2; 1; 1) \quad . \tag{29}$$

It turns out that ${}_2F_1(-l, l+1/2; 1; 1) = (1/2-l)_l/l!$, where $(k)_l$ is a Pochammer coefficient. Some more lines of calculus give $(1/2-l)_l = (-1)^l 2^{-2l}(2l)!/l!$. This produces the expected result. Secondly,

$$\frac{\beta_{l+1}}{\beta_l} = \left(\frac{l+1/2}{l+1}\right)^2 < 1 \quad . \tag{30}$$

References

1. Tardos, G.: Optimal probabilistic fingerprint codes. In: Proc. of the 35th annual ACM symposium on theory of computing, San Diego, CA, USA, ACM (2003) 116–125
2. Skoric, B., Katzenbeisser, S., Celik, M.: Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs, Codes and Cryptography **46** (2008) 137–166
3. Skoric, B., Vladimirova, T., Celik, M., Talstra, J.: Tardos fingerprinting is better than we thought. arXiv:cs/0607131v1 (2006)
4. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimal probabilistic fingerprinting codes using optimal finite random variables related to numerical quadrature. arXiv:cs/0610036v2 (2006)
5. Katzenbeisser, S., Skoric, B., Celik, M., Sadeghi, A.R.: Combining Tardos fingerprinting codes and fingercasting. In Verlag, S., ed.: Proc. of 9th Information Hiding. Volume 4567 of Lecture Notes in Computer Science. (2007)
6. Barg, A., Blakley, G.R., Kabatiansky, G.A.: Digital fingerprinting codes: problem statements, constructions, identification of traitors. IEEE Trans. Inform Theory **49** (2003) 852–865
7. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory **44** (1998) 1897–1905
8. Safavi-Naini, R., Wang, Y.: Collusion-secure q-ary fingerprinting for perceptual content. In Springer-Verlag, ed.: Proc. Security and Privacy in Digital Rights Management, SPDRM'01. Volume 2320 of Lecture Notes in Computer Science. (2001) 57–75