The Holey Grail : a special score function for non-binary traitor tracing

The Holey Grail
A special score function for non-binary traitor tracing

B. Škorić, J.-J. Oosterwijk, J. Doumen

***Abstract*—We study collusion-resistant traitor tracing in the simple decoder approach, i.e. assignment of scores for each user separately. We introduce a new score function for non-binary bias-based traitor tracing. It has three special properties that have long been sought after: (i) The expected score of an innocent user is zero in each content position. (ii) The variance of an innocent user's score is 1 in each content position. (iii) The expectation of the coalition's score does not depend on the collusion strategy.**

We also find a continuous bias distribution that optimizes the asymptotic (large coalition) performance. In the case of a binary alphabet our scheme reduces exactly to the symmetrized Tardos traitor tracing system. Unfortunately, the asymptotic fingerprinting rate of our new scheme decreases with growing alphabet size. We regret to inform you that this grail has holes.

I. Introduction

A. Collusion-resistant tracing

Forensic watermarking is a means for tracing the origin and (re-)distribution of digital content. Before distribution, the content is modified by embedding an imperceptible watermark, which plays the role of a personalized serial number. When an unauthorized copy of the content is found, the identities of those users who participated in its creation can be determined from the watermark. A tracing algorithm outputs a list of suspicious users.

Collusion attacks are a powerful class of attacks against forensic watermarking. Multiple attackers (referred to as *colluders* or a *coalition*) combine their differently watermarked versions of the same content. The observed differences point to the locations of the hidden marks. Knowledge of these locations helps the colluders to mix and match their versions.

Different types of collusion-resistant codes have been developed in order to defend against these attacks. The most popular in the recent literature is the class of *bias-based* codes. These were introduced by G. Tardos in 2003. The original paper [1] was followed by a lot of activity, e.g. improved analyses [2], [3], [4], [5], [6], [7], code modifications [8], [9], [10], decoder modifications [11], [12], [13] and various generalizations [14], [15], [16], [17]. The advantage of bias-based versus deterministic codes is that they can achieve the asymptotically optimal relationship $\ell \propto c^2$ between the sufficient code length ℓ and the coalition size c.

We distinguish between two kinds of tracing algorithm: (i) *simple decoders*, which assign a score to individual users and (ii) *joint decoders* [11], [12], [13], which look at sets of users. Joint decoders sometimes employ a simple decoder as a bootstrapping step. Tardos' original scheme [1] and its symmetrized version [15] work with a simple decoder. The Amiri-Tardos accusation scheme [11] and the Don Quixote scheme [13] are examples of joint decoders.

In the study of collusion-resistant watermarking one often uses a model in which the details of the watermark embedding process have been abstracted away. The content is considered to consist of a number of 'segments', 'positions' or 'locations' into each of which a symbol from an alphabet $\mathcal{Q}$ can be embedded. A position in which not all the colluders have received the same symbol is called a *detectable position*. It is customary to assume that the so-called Marking Assumption holds: the colluders are able to modify the watermark only in detectable positions. Furthermore, one often adopts the Restricted Digit Model (RDM) as attacker model because of its simplicity and amenability to analysis. The RDM states that the attackers may only output a symbol from the set of symbols they received (and not for instance an erasure, a different symbol, or a mixture of multiple symbols).

Most of the literature on content tracing works with a binary alphabet. However, it has been shown that larger alphabets can offer a higher fingerprinting rate: in the RDM the fingerprinting capacity in the large c limit is given [18] by $C_q = (q-1)/(2c^2 \ln q)$, where $q = |\mathcal{Q}|$ is the alphabet size.

In this paper we focus on score functions for simple-decoder bias-based tracing in the case of arbitrary-size coalitions and non-binary alphabets. We work in the Restricted Digit Model.

B. Related work

The symmetrized version of Tardos' original score function for $q = 2$ has asymptotic (large c) fingerprinting rate $2/(c^2\pi^2 \ln 2)$, which is roughly a factor 2.5 below capacity. Its generalization [15] to $q \geq 3$ outperforms [19] the binary scheme but is far below the q-ary capacity.

Amiri and Tardos [11] devised a capacity-achieving joint decoder for $q = 2$. Unfortunately, it is computationally intensive since it requires looking at all candidate coalitions. A non-binary version has not yet been described, though generalization seems straightforward.

The 'Divide and Conquer' scheme for q-ary alphabets, introduced by Laarhoven et al. [20], works in the *dynamic* setting (e.g. pay-TV broadcast), where the content tracer immediately sees the result of the attack on a position and uses this information to decide which symbols to distribute in the next position. This approach intertwines several 'ordinary' Tardos schemes of lower alphabet size. Its asymptotic fingerprinting rate is $\frac{2}{\pi^2}\frac{q}{q-1}C_q$ when instantiated with the symmetric Tardos score. In this paper we will not consider the dynamic setting.

There are several studies of bias-based fingerprinting in attack models that deviate from the Marking Assumption and the RDM. Some of these introduce modified simple-decoder score functions. For instance, one can allow noise addition and fusion of symbols; modified score functions were proposed and analyzed in [16], [17].

Kuribayashi [21] introduced a score function modification for the binary case that aims to exploit imbalances between the 0s and 1s in the attacked content.

The Expectation Maximization (EM) algorithm [12] was introduced as an iterative joint decoder. It estimates a candidate coalition. Based on this set of users it estimates the employed collusion strategy. Then the simple-decoder score function is modified to act specifically against this collusion strategy. The scores are used to find suspicious users, and the whole procedure is repeated. For $q = 2$ a formula was given [12] for computing a score function optimized against an estimated strategy.

This was extended to arbitrary alphabet size by Oosterwijk et al. [22], and furthermore analytic expressions were obtained for the score functions optimized against the Interleaving, Majority Voting, Minority Voting, Random Symbol and All-High strategy. The score function tailored against Interleaving ('Interleaving defense') is special. For this score it was shown [23] that the saddlepoint of a minimax game between the coalition and the tracer is given by the same configuration that was found by Huang and Moulin [24] for the capacity game: the Interleaving attack combined with the Dirichlet bias distribution (with concentration parameter $\frac{1}{2}$). At the saddle point the asymptotic fingerprinting rate achieved by the Interleaving defense is exactly C_q. In other words: (i) for large c there is no better simple-decoder scheme than the Interleaving defense together with the Dirichlet bias distribution; (ii) the most powerful attack against this scheme is the Interleaving strategy; (iii) the scheme achieves asymptotic capacity.

C. Contributions

We introduce a special score function for non-binary bias-based fingerprinting. It has three interesting properties:

1) The expected score of an innocent user is zero.
2) The variance of an innocent user's score is one.
3) The expectation of the colluders' summed scores does not depend on the collusion strategy.

We also find a continuous bias distribution that optimizes the asymptotic performance of the scheme. In the case of a binary alphabet our scheme reduces exactly to the symmetrized Tardos fingerprinting scheme.

The combination of the above three simplifying properties, exhibited by the *binary* Tardos scheme, has been long sought after for $q > 2$ and is regarded as something of a holy grail. Unfortunately the asymptotic performance of the grail is not good. The asymptotic fingerprinting rate is a *decreasing* function of q. Thus it performs worse than the q-ary scheme of Škorić et al. [15], which has a rate that is almost constant at approximately $0.3/c^2$ [19], and far worse than the capacity-achieving Interleaving defense [23].

In this light our newly found scheme is somewhat of an anticlimax. After a long quest for a bias-based q-ary scheme with precisely the same special properties as the symmetrized binary Tardos scheme, the prize seems to be a mere curiosity. Perhaps it has a role to play at small coalition sizes.

II. Preliminaries

A. Code construction, collusion attack, and simple decoder

We briefly summarize the basics of bias-based codes ('Tardos codes') in the Restricted Digit Model and the notation used in this paper.

The number of users is n. The set $\{1, \ldots, n\}$ is denoted as $[n]$. The content has ℓ positions in which a symbol can be embedded. The symbols belong to an alphabet $\mathcal{Q}$, with size $q = |\mathcal{Q}| \geq 2$. In each position, the tracer draws a random q-component bias vector $\boldsymbol{p}$ from a distribution F, with $\boldsymbol{p} \in [0,1]^{\mathcal{Q}}$, $\boldsymbol{p} \sim F$. The components are denoted as $p_\alpha \in [0,1]$, i.e. $\boldsymbol{p} = (p_\alpha)_{\alpha \in \mathcal{Q}}$. The bias vector satisfies $\sum_{\alpha \in \mathcal{Q}} p_\alpha = 1$. The tracer uses the bias vector to generate code symbols for the given position as follows. Let X_j denote the symbol given to user j. The tracer generates symbols randomly according to $\mathrm{Prob}[X_j = \alpha] = p_\alpha$.

The coalition is a set of users $\mathcal{C} \subset [n]$. They observe a subset of X, which we denote as $X_\mathcal{C}$. They perform their attack based on $X_\mathcal{C}$. In the Restricted Digit Model, they are allowed to choose, in each position, one symbol that they observed in that position. Their output symbol is denoted as y. Their strategy for choosing y may be nondeterministic. We will use the notation $\theta_{y|X_\mathcal{C}}$ to denote their probability of outputting y given $X_\mathcal{C}$. We refer to the parameters $\theta_{y|X_\mathcal{C}}$ as the 'strategy' or the 'attack'.

The tracer tries to identify at least one of the colluders, based on the information available to him: the $\boldsymbol{p}$, X, and y values in all the positions. We consider a class of algorithms known as 'simple decoder', in which a score is assigned to each user $j \in [n]$ separately. More specifically, we consider single-position contributions S_j that are added up. If the sum exceeds some threshold, user j is 'accused'. The maximum tolerable probability that a fixed innocent user gets accused is denoted as ε_1.

In the decoder that we consider, the single-position scores are computed as

$$S_j = h(X_j, y, \boldsymbol{p}), \tag{1}$$

where h is some function and the position index on S_j, X_j, y and $\boldsymbol{p}$ is omitted. Without loss of generality, we will consider only score functions h such that the expectation value of an innocent user's score is zero. We call such score functions *centered*. (One can shift a non-centered h by a constant to make it centered, without changing the properties of the scheme at all.)

The generalized (q-ary) Tardos scheme [15] has

$$S_j = \begin{cases} g_1(p_y) & \text{if } X_j = y \\ g_0(p_y) & \text{if } X_j \neq y \end{cases} \tag{2}$$

where

$$g_1(p) = \sqrt{\frac{1-p}{p}} \qquad ; \qquad g_0(p) = -\sqrt{\frac{p}{1-p}}. \tag{3}$$

B. Asymptotic analysis

We focus on the asymptotic (large c, with n/c fixed) analysis of the bias-based tracing scheme. We will need to compute expectation values over *all* probabilistic degrees of freedom: the biases $\boldsymbol{p}$, the code word symbols X, and the coalition outputs y. The notation for the complete expectation will be $\mathbb{E}$, whereas expectation with respect to $\boldsymbol{p}$ has notation $\mathbb{E}_{\boldsymbol{p}}$ etc.
A noteworthy variable is the *tally* of symbols received by the coalition. We define $m_\alpha = |\{j \in \mathcal{C} : X_j = \alpha\}|$. In words: m_α counts how many colluders have received symbol α. In each position the tally adds up to c: we have $\sum_{\alpha\in\mathcal{Q}} m_\alpha = c$. The vector $\boldsymbol{m} = (m_\alpha)_{\alpha\in\mathcal{Q}}$ given $\boldsymbol{p}$ is multinomial-distributed,

$$P_{\boldsymbol{m}|\boldsymbol{p}} = \binom{c}{\boldsymbol{m}} \prod_{\alpha\in\mathcal{Q}} p_\alpha^{m_\alpha}. \tag{4}$$

Here $\binom{c}{\boldsymbol{m}}$ stands for the multinomial coefficient $c!/\prod_\alpha m_\alpha!$. We will often use the multi-index notation $\boldsymbol{p}^{\boldsymbol{m}} = \prod_{\alpha\in\mathcal{Q}} p_\alpha^{m_\alpha}$ and for a scalar s, $\boldsymbol{p}^s = \prod_\alpha p_\alpha^s$.
The collective coalition score $S_\mathcal{C}$ in a certain position is defined as

$$S_\mathcal{C} = \sum_{j\in\mathcal{C}} S_j = \sum_{\alpha\in\mathcal{Q}} m_\alpha h(\alpha, y, \boldsymbol{p}). \tag{5}$$

Two important statistical quantities were introduced [7]: the expectation $\tilde{\mu}_c$ of the coalition score and the variance $\tilde{\sigma}^2_{\rm inn}$ of an innocent's score. The first one is given by

$$\begin{aligned} \tilde{\mu}_c &= \mathbb{E}[S_\mathcal{C}] = \sum_{\alpha\in\mathcal{Q}} \mathbb{E}[m_\alpha h(\alpha, y, \boldsymbol{p})] \\ &= \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{\boldsymbol{m}|\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{m}}[\sum_{\alpha\in\mathcal{Q}} m_\alpha h(\alpha, y, \boldsymbol{p})]. \end{aligned} \tag{6}$$

Remark 1: $\tilde{\mu}_c$ may depend on the (omitted) position index i; this happens when the attack strategy has explicit position-dependence, breaking the symmetry that is present in the code generation and tracing algorithms.
Remark 2: The single-position quantity $\mathbb{E}_{y|\boldsymbol{m}}$ is the marginal of complicated expectations involving $\boldsymbol{p}$ and $\boldsymbol{m}$ vectors in all other positions as well as the attack strategy that possibly depends on *all* positions. We have used that, for any single-position function $f(\boldsymbol{p}, \boldsymbol{m}, y)$, one can write $\mathbb{E}[f(\boldsymbol{p}, \boldsymbol{m}, y)] = \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{\boldsymbol{m}|\boldsymbol{p}}\mathbb{E}_{X_\mathcal{C}|\boldsymbol{p}\boldsymbol{m}}\mathbb{E}_{y|X_\mathcal{C}}[f(\boldsymbol{p}, \boldsymbol{m}, y)]$. Now the probability of $X_\mathcal{C}$ occurring given $\boldsymbol{p}$ satisfies $P_{X_\mathcal{C}|\boldsymbol{p}} \propto \prod_{j\in\mathcal{C}} p_{X_j} = \boldsymbol{p}^{\boldsymbol{m}} \propto P_{\boldsymbol{m}|\boldsymbol{p}}$. In other words, all the $\boldsymbol{p}$-dependence in $P_{X_\mathcal{C}|\boldsymbol{p}}$ is already contained in $P_{\boldsymbol{m}|\boldsymbol{p}}$. Hence for given $\boldsymbol{m}$, the distribution of $X_\mathcal{C}$ has no extra dependence on $\boldsymbol{p}$, which allows us to write $\mathbb{E}_{X_\mathcal{C}|\boldsymbol{p}\boldsymbol{m}} = \mathbb{E}_{X_\mathcal{C}|\boldsymbol{m}}$ and therefore $\mathbb{E}_{X_\mathcal{C}|\boldsymbol{p}\boldsymbol{m}}\mathbb{E}_{y|X_\mathcal{C}} = \mathbb{E}_{y|\boldsymbol{m}}$, yielding (6).
The second statistical quantity is, for $j \notin \mathcal{C}$ and a centered score function,

$$\begin{aligned} \tilde{\sigma}^2_{\rm inn} &= \mathbb{E}[S_j^2] = \mathbb{E}[h^2(X_j, y, \boldsymbol{p})] \\ &= \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{p}}\mathbb{E}_{X_j|\boldsymbol{p}}[h^2(X_j, y, \boldsymbol{p})] \\ &= \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{p}}[\sum_{x\in\mathcal{Q}} p_x h^2(x, y, \boldsymbol{p})]. \end{aligned} \tag{7}$$

In the first line we have used that the expectation of S_j is zero. In the second line we have used that $X_j|\boldsymbol{p}$ and $y|\boldsymbol{p}$ are independent for an innocent user j, and in the third line that $P_{X_j|\boldsymbol{p}} = p_{X_j}$. Note that $P_{y|\boldsymbol{p}}$ can be extremely complicated, containing expectations over all bias vectors, coalition symbols and coalition outputs *in other positions*.
For any function $z(\boldsymbol{p})$ we have

$$\mathbb{E}_{\boldsymbol{p}}[z(\boldsymbol{p})] = \int_0^1 \mathrm{d}^q p\ \delta(1 - \sum_{\alpha\in\mathcal{Q}} p_\alpha) F(\boldsymbol{p}) z(\boldsymbol{p}). \tag{8}$$

Here the notation $\int \mathrm{d}^q p$ stands for integration over the hypercube $\boldsymbol{p} \in [0,1]^\mathcal{Q}$, and the Dirac delta function $\delta(1 - \sum_\alpha p_\alpha)$ enforces the constraint $\sum_\alpha p_\alpha = 1$.
Further on we will encounter *Dirichlet integrals*, also known as generalized Beta functions. Let $\boldsymbol{v} \in (0,\infty)^q$ be a vector, then

$$\int_0^1 \mathrm{d}^q p\ \delta(1 - \sum_\alpha p_\alpha) \boldsymbol{p}^{-1+\boldsymbol{v}} = B(\boldsymbol{v}) = \frac{\prod_\beta \Gamma(v_\beta)}{\Gamma(\sum_\alpha v_\alpha)}. \tag{9}$$

Here Γ is the Gamma function, with the property $\Gamma(1+x) = x\Gamma(x)$.
Asymptotically the performance of the simple-decoder tracing scheme as described above depends on a single parameter, namely the fraction $\tilde{\mu}_c/\tilde{\sigma}_{\rm inn}$ [7]. Asymptotically, the sufficient code length ℓ and the fingerprinting rate R are given by

$$\ell = \frac{2\tilde{\sigma}^2_{\rm inn}}{\tilde{\mu}_c^2} c^2 \ln\frac{1}{\varepsilon_1} \qquad ; \qquad R = \frac{\tilde{\mu}_c^2}{c^2 \cdot 2\tilde{\sigma}^2_{\rm inn} \ln q}. \tag{10}$$

Here it is implicit that $\tilde{\mu}_c/\tilde{\sigma}_{\rm inn}$ is averaged over all positions if necessary. (Which is only the case for symmetry-breaking strategies).

III. A new score function and bias distribution

The main contribution of this paper is the introduction of a new simple-decoder score function for q-ary fingerprinting,

$$h(x, y, \boldsymbol{p}) = \frac{a_q[F]}{F(\boldsymbol{p})}\left[\frac{(-1)^{1+\delta_{xy}}}{p_x} + q - 2\right] \tag{11}$$

$$a_q[F] = \left(\mathbb{E}_{\boldsymbol{p}} \frac{1}{F^2(\boldsymbol{p})}[\sum_{\alpha\in\mathcal{Q}} \frac{1}{p_\alpha} - (q-2)^2]\right)^{-1/2} \tag{12}$$

Here δ_{xy} is a Kronecker delta; the $a_q[F]$ is a (positive) F-dependent normalization constant that makes sure that $\tilde{\sigma}^2_{\rm inn} = 1$ and that the symmetric score function (2) is re-obtained at $q = 2$.

The score (11) has the following properties, which hold for any bias distribution F,

- An innocent user's score has expectation value zero.
- The variance of an innocent user's score is one.
- The expectation value of the coalition score does not depend on the collusion strategy.

Furthermore, we find that the following bias distribution maximizes the performance indicator $\tilde{\mu}_c/\tilde{\sigma}_{\rm inn}$,

$$F(\boldsymbol{p}) = \frac{1}{\mathcal{N}_q}\sqrt{\sum_{\alpha\in\mathcal{Q}}\frac{1}{p_\alpha}-(q-2)^2} \quad (13)$$

$$\mathcal{N}_q = \int_0^1 \mathrm{d}^q p\ \delta(1-\sum_{\beta\in\mathcal{Q}} p_\beta)\sqrt{\sum_{\alpha\in\mathcal{Q}}\frac{1}{p_\alpha}-(q-2)^2} \quad (14)$$

where $\mathcal{N}_q$ is a normalization constant. With this choice of F, the normalization constant becomes $a_q[F]=1/\mathcal{N}_q$.
Eqs. (11) and (13) together form a 'cleaner' generalization of the symmetric binary score system to q-ary alpabets than the earlier scheme [15], in the sense that it preserves more of the strategy-independence properties. Below we prove all the above mentioned claims one by one.

A. Properties of the score function

Definition 1 (Strongly centered): A score function $h(x,y,\boldsymbol{p})$ is called strongly centered if it satisfies $\sum_{x\in\mathcal{Q}} p_x h(x,y,\boldsymbol{p}) = 0$.

Theorem 1: The score function (11) is strongly centered.
Proof: The sum $\sum_{x\in\mathcal{Q}} p_x h(x,y,\boldsymbol{p})$ is proportional to

$$\sum_{x\in\mathcal{Q}} p_x[\frac{(-1)^{1+\delta_{xy}}}{p_x}+q-2] = \sum_{x\in\mathcal{Q}}(-1)^{1+\delta_{xy}}+q-2 = 0. \quad (15)$$

In the first equality we used $\sum_x p_x = 1$. In the last step we used $\sum_x(-1)^{1+\delta_{xy}} = 2-q$. □

Theorem 2: If the score function (11) is used, the variance of an innocent user's score is equal to one.
Proof: Eq. (7) evaluates to

$$\begin{aligned}\frac{\tilde{\sigma}^2_{\rm inn}}{a_q^2[F]} &= \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{p}}\sum_{x\in\mathcal{Q}} p_x\frac{1}{F^2(\boldsymbol{p})}\left[(q-2)^2\right.\\ &\qquad \left.+\frac{1}{p_x^2}+2(q-2)(-1)^{1+\delta_{xy}}\frac{1}{p_x}\right]\\ &= \mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{p}}\frac{1}{F^2(\boldsymbol{p})}\left[\sum_{x\in\mathcal{Q}}\frac{1}{p_x}-(q-2)^2\right]. \end{aligned} \quad (16)$$

In the last line we have used $\sum_x p_x = 1$ and $\sum_x(-1)^{1+\delta_{xy}} = 2-q$. The expression between brackets in (16) does not depend on y; hence the expectation $\mathbb{E}_{y|\boldsymbol{p}}$ is trivial and (16) reduces to a_q^{-2}, with a_q as defined by (12). □

Theorem 3: The score function (11) gives

$$\tilde{\mu}_c = \frac{2a_q[F]}{(q-2)!} \quad (17)$$

independent of the colluder strategy.

Proof: We write

$$\sum_{\alpha\in\mathcal{Q}} m_\alpha h(\alpha,y,\boldsymbol{p}) = \frac{a_q[F]}{F(\boldsymbol{p})}[c(q-2)+2\frac{m_y}{p_y}-\sum_{\alpha\in\mathcal{Q}}\frac{m_\alpha}{p_\alpha}] \quad (18)$$

and substitute this into (6). The expectation of the first term is $T_1 := a_q[F]c(q-2)\mathbb{E}_{\boldsymbol{p}}\frac{1}{F(\boldsymbol{p})}$. For the expectation of the third term in (18) we use the fact that $\mathbb{E}_{\boldsymbol{m}|\boldsymbol{p}} m_\alpha = cp_\alpha$ and obtain $T_3 := -a_q[F]qc\mathbb{E}_{\boldsymbol{p}}\frac{1}{F(\boldsymbol{p})}$. The second term in (18) is more difficult. Here we get

$$\begin{aligned} T_2 &:= 2a_q[F]\mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{\boldsymbol{m}|\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{m}}\frac{m_y}{p_y F(\boldsymbol{p})}\\ &= 2a_q[F]\sum_{\boldsymbol{m}}\binom{c}{\boldsymbol{m}}\mathbb{E}_{y|\boldsymbol{m}}\mathbb{E}_{\boldsymbol{p}}\frac{m_y\boldsymbol{p}^{\boldsymbol{m}}}{p_y F(\boldsymbol{p})}. \end{aligned} \quad (19)$$

Here it is implicit that all the $\boldsymbol{m}$-vectors in the summation satisfy $\sum_\alpha m_\alpha = c$. The $\mathbb{E}_{\boldsymbol{p}}$ is computed as follows,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{p}}\frac{m_y\boldsymbol{p}^{\boldsymbol{m}}}{p_y F(\boldsymbol{p})} &= m_y\int_0^1 \mathrm{d}^q p\ \delta(1-\sum_\beta p_\beta)\frac{\boldsymbol{p}^{\boldsymbol{m}}}{p_y}\\ &= m_y B(\mathbf{1}_q+\boldsymbol{m}-\boldsymbol{e}_y) = (c+q-1)B(\mathbf{1}_q+\boldsymbol{m})\\ &= (c+q-1)\mathbb{E}_{\boldsymbol{p}}\frac{\boldsymbol{p}^{\boldsymbol{m}}}{F(\boldsymbol{p})}. \end{aligned} \quad (20)$$

Here $\mathbf{1}_q$ denotes the q-component vector $(1,1,\ldots,1)$, and $\boldsymbol{e}_y$ is a q-component vector consisting of all zeroes except in position y, i.e. $(\boldsymbol{e}_y)_\alpha = \delta_{\alpha y}$. Substitution of (20) into (19) gives

$$\begin{aligned} T_2 &= 2a_q[F](c+q-1)\mathbb{E}_{\boldsymbol{p}}\mathbb{E}_{\boldsymbol{m}|\boldsymbol{p}}\mathbb{E}_{y|\boldsymbol{m}}\frac{1}{F(\boldsymbol{p})}\\ &= 2a_q[F](c+q-1)\mathbb{E}_{\boldsymbol{p}}\frac{1}{F(\boldsymbol{p})}. \end{aligned} \quad (21)$$

In the last equality we have used that $1/F(\boldsymbol{p})$ does not depend on $\boldsymbol{m}$ and y. Adding $T_1+T_2+T_3$ we get

$$\begin{aligned}\frac{\tilde{\mu}_c}{2a_q[F]} &= (q-1)\mathbb{E}_{\boldsymbol{p}}\frac{1}{F(\boldsymbol{p})} = (q-1)\int_0^1 \mathrm{d}^q p\ \delta(1-\sum_\beta p_\beta)\\ &= (q-1)B(\mathbf{1}_q) = \frac{(q-1)}{\Gamma(q)} = \frac{1}{(q-2)!}. \end{aligned} \quad (22)$$

□

Note that in the expression $\mathbb{E}_{\boldsymbol{p}}[\boldsymbol{p}^{\boldsymbol{m}}/(p_y F)]$, the factor p_y^{-1} does not pose a problem, because $m_y \geq 1$ in the Restricted Digit Model. In contrast, $\mathbb{E}_{\boldsymbol{p}}[\boldsymbol{p}^{\boldsymbol{m}}/(p_\alpha F)]$ for $\alpha \neq y$ does not always exist: the integral may be divergent when $m_\alpha = 0$. For this reason, in the proof of Theorem 3 we avoided the expression $\mathbb{E}_{\boldsymbol{p}}[\boldsymbol{p}^{\boldsymbol{m}}/(p_\alpha F)]$ when the third term of (18) was averaged.

B. Optimal bias distribution F

Theorem 4: The performance indicator $\tilde{\mu}_c/\tilde{\sigma}_{\rm inn}$ is maximized by the bias distribution (13).
Proof: The $\tilde{\sigma}_{\rm inn}$ is equal to 1. We minimize $\tilde{\mu}_c^{-2}$ under the constraint $\mathbb{E}_{\boldsymbol{p}}[1] = 1$ using the Euler-Lagrange method. From (17) we see that this is equivalent to minimizing $(a_q[F])^{-2}$. The corresponding Lagrangian can be formulated

as $a_q^{-2}[F] + \lambda[\int_0^1 \mathrm{d}^q p\ \delta(1-\sum_\alpha p_\alpha)F(\boldsymbol{p}) - 1]$, with $a_q^{-2}[F]$ being the $\mathbb{E}_{\boldsymbol{p}}$-integral defined in (12). Here λ is a Lagrange multiplier. Functional differentiation of the Lagrangian with respect to $F(\boldsymbol{p})$ gives $0 = \lambda - \frac{1}{F^2(\boldsymbol{p})}[\sum_\alpha \frac{1}{p_\alpha} - (q-2)^2]$. Solving for F, and respecting the normalization constraint, yields (13). □

Theorem 5: For $q = 2$, the combination of the score function (11) with the bias distribution (13) reproduces the binary symmetric scheme of [15] with zero cutoff.

Proof: For $q = 2$ the bias function (13) is $(1/\mathcal{N}_2)\sqrt{\sum_\alpha p_\alpha^{-1}} = (1/\mathcal{N}_2)(\prod_\alpha p_\alpha)^{-1/2}$ and the normalization constant reduces to $\mathcal{N}_2 = \pi$. This is precisely the arcsine distribution $f(p) = (1/\pi)[p(1-p)]^{-1/2}$ as introduced by Tardos [1].
For $q = 2$ Eq. (11) gives $h(y, y, \boldsymbol{p}) = \sqrt{(1-p_y)/p_y}$, and for $x \neq y$: $h(x, y, \boldsymbol{p}) = -\sqrt{(1-p_x)/p_x} = -\sqrt{p_y/(1-p_y)}$. This is the old score system (2). □

C. Asymptotic performance

Corollary 1: The asymptotic code length ℓ and asymptotic fingerprinting rate R of the new scheme are

$$\ell = A_q c^2 \ln \frac{1}{\varepsilon_1}, \quad R = \frac{1}{A_q c^2 \ln q}, \tag{23}$$

$$\text{with } A_q = \tfrac{1}{2}[(q-2)!]^2 \mathcal{N}_q^2. \tag{24}$$

Proof: Follows by substituting Theorems 3 and 4 into (10). □
Numerical values for $\mathcal{N}_q$ are tabulated below for $q \leq 13$.

q	2	3	4	5	6	7	8
$\mathcal{N}_q$	π	2.65	1.24	0.401	9.88E-2	1.96E-2	3.26E-3

q	9	10	11	12	13
$\mathcal{N}_q$	4.65E-4	5.82E-5	6.47E-6	6.50E-7	5.93E-8

The asymptotic code length parameter A_q and the asymptotic fingerprinting rate parameter $1/(A_q \ln q)$ are plotted in Fig. 1. The rate parameter *decreases* as a function of q, whereas the fingerprinting capacity *increases*.

IV. Summary and discussion

Summarizing, we have introduced a q-ary generalization of the binary symmetrized Tardos scheme which preserves the strategy-independent properties of the binary scheme. The bias distribution is given by (14) and the generalization of the score function (2) is

$$h(x, y, \boldsymbol{p}) = \frac{\frac{(-1)^{1+\delta_{xy}}}{p_x} + q - 2}{\sqrt{\sum_\alpha \frac{1}{p_\alpha} - (q-2)^2}}. \tag{25}$$

This combination of bias distribution and score function yields $\tilde{\sigma}^2_{\mathrm{inn}} = 1$ and $\tilde{\mu}_c = \frac{2}{\mathcal{N}_q (q-2)!}$, with $\mathcal{N}_q$ as defined in (14).
In spite of all the nice properties, it turns out that, as far as we can see from the numerics, the asymptotic fingerprinting rate is a *decreasing* function of the alphabet size q; the new scheme performs worse than other q-ary schemes known in the literature.
The analysis in this paper is brief and focuses on large-c asymptotics. In spite of its bad asymptotic performance, our newly found scheme may have a role to play at small coalition sizes.

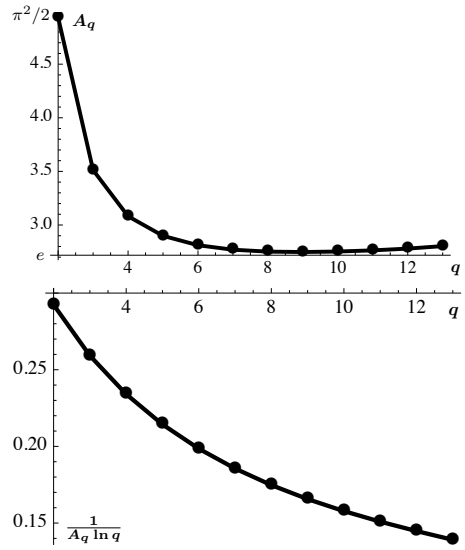

Fig. 1. **Top:** *The asymptotic code length parameter A_q as a function of q.* **Bottom:** *The asymptotic fingerprinting rate parameter $1/(A_q \ln q)$ as a function of q.*

Acknowledgements

We thank Thijs Laarhoven and Benne de Weger for useful discussions. We thank Wil Kortsmit for his help with the numerics. Part of this research was funded by STW, project number 10518.

References

[1] G. Tardos, "Optimal probabilistic fingerprint codes," in *ACM Symposium on Theory of Computing (STOC) 2003*, pp. 116–125.
[2] O. Blayer and T. Tassa, "Improved versions of Tardos' fingerprinting scheme," *Des. Codes Cryptogr.*, vol. 48, no. 1, pp. 79–103, 2008.
[3] T. Furon, A. Guyader, and F. Cérou, "On the design and optimization of Tardos probabilistic fingerprinting codes," in *Information Hiding 2008*, ser. LNCS, vol. 5284. Springer, pp. 341–356.
[4] T. Furon, L. Pérez-Freire, A. Guyader, and F. Cérou, "Estimating the minimal length of Tardos code," in *Information Hiding 2009*, ser. LNCS, vol. 5806. Springer, pp. 176–190.
[5] T. Laarhoven and B. de Weger, "Optimal symmetric Tardos traitor tracing schemes," *Des. Codes Cryptogr.*, 2012, http://arxiv.org/abs/1107.3441.
[6] A. Simone and B. Škorić, "Accusation probabilities in Tardos codes," *Benelux Workshop on Information and System Security (WISSEC) 2010.* Preprint available at http://eprint.iacr.org/2010/472.
[7] B. Škorić, T. Vladimirova, M. Celik, and J. Talstra, "Tardos fingerprinting is better than we thought," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3663–3676, 2008.
[8] Y.-W. Huang and P. Moulin, "Capacity-achieving fingerprint decoding," in *IEEE Workshop on Information Forensics and Security*, 2009, pp. 51–55.
[9] K. Nuida, "Short collusion-secure fingerprint codes against three pirates," in *Information Hiding 2010*, ser. LNCS, vol. 6387. Springer, pp. 86–102.
[10] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of discrete Tardos fingerprinting codes," *Des. Codes Cryptography*, vol. 52, no. 3, pp. 339–362, 2009.

[11] E. Amiri and G. Tardos, "High rate fingerprinting codes and the fingerprinting capacity," in *ACM-SIAM Symposium On Discrete Algorithms (SODA) 2009*, pp. 336–345.
[12] A. Charpentier, F. Xie, C. Fontaine, and T. Furon, "Expectation maximization decoding of Tardos probabilistic fingerprinting code," in *Media Forensics and Security 2009*, ser. SPIE Proceedings, vol. 7254, p. 72540.
[13] P. Meerwald and T. Furon, "Towards Joint Tardos Decoding: The 'Don Quixote' Algorithm," in *Information Hiding 2011*, ser. LNCS, vol. 6958. Springer, pp. 28–42.
[14] A. Charpentier, C. Fontaine, T. Furon, and I. Cox, "An asymmetric fingerprinting scheme based on Tardos codes," in *Information Hiding*, ser. LNCS, vol. 6958. Springer, 2011, pp. 43–58.
[15] B. Škorić, S. Katzenbeisser, and M. Celik, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Des. Codes Cryptogr.*, vol. 46, no. 2, pp. 137–166, 2008.
[16] B. Škorić, S. Katzenbeisser, H. Schaathun, and M. Celik, "Tardos fingerprinting codes in the Combined Digit Model," in *IEEE Workshop on Information Forensics and Security (WIFS) 2009*, pp. 41–45.
[17] F. Xie, T. Furon, and C. Fontaine, "On-off keying modulation and Tardos fingerprinting," in *ACM Workshop on Multimedia and Security (MM&Sec) 2008*, pp. 101–106.
[18] D. Boesten and B. Škorić, "Asymptotic fingerprinting capacity for non-binary alphabets," in *Information Hiding 2011*, ser. LNCS, vol. 6958. Springer, pp. 1–13.
[19] B. Škorić and J.-J. Oosterwijk, "Binary and q-ary tardos codes, revisited," http://eprint.iacr.org/2012/249.
[20] T. Laarhoven, J.-J. Oosterwijk, and J. Doumen, "Dynamic traitor tracing for arbitrary alphabets: Divide and conquer," in *IEEE Workshop on Information Forensics and Security (WIFS) 2012*, pp. 240–245.
[21] M. Kuribayashi, "Bias equalizer for binary probabilistic fingerprinting codes," in *Information Hiding 2012*, ser. LNCS, vol. 7692, pp. 269–283.
[22] J.-J. Oosterwijk, B. Škorić, and J. Doumen, "Optimal suspicion functions for Tardos traitor tracing schemes," in *ACM Workshop on Information Hiding and Multimedia Security 2013*, pp. 19–28.
[23] J.-J. Oosterwijk, B. Škorić, and J. Doumen, "A capacity-achieving simple decoder for bias-based traitor tracing schemes," http://eprint.iacr.org/2013/389.
[24] Y.-W. Huang and P. Moulin, "On fingerprinting capacity games for arbitrary alphabets and their asymptotics," *IEEE International Symposium on Information Theory (ISIT) 2012*, pp. 2571–2575.