Generalised tally-based decoders for traitor tracing and group testing

Boris Škorić and Wouter de Groot

***Abstract*—We propose a new type of score function for Tardos traitor tracing codes. It is related to the recently introduced tally-based score function [27], but it utilizes more of the information available to the decoder. It does this by keeping track of *sequences of symbols* in the distributed codewords instead of looking at columns of the code matrix individually.**
We derive our new class of score functions from a Neyman-Pearson hypothesis test and illustrate its performance with simulation results.
Finally we derive a score function for (medical) group testing applications.

I. Collusion attacks on watermarking

Forensic watermarking is a means for tracing the origin and distribution of digital content. Before distribution, the content is modified by embedding an imperceptible watermark, which is unique for each recipient. When an unauthorized copy of the content is found, the watermark present in this copy reveals information about the identities of those who created the copy. A tracing algorithm ('decoder') outputs a list of suspicious users. This procedure is known as Traitor Tracing.

The most powerful attacks against watermarks are *collusion attacks*: multiple attackers (the 'coalition') combine their differently watermarked versions of the same content; the observed differences point to the locations of the hidden marks and allow for an informed attack.

Several types of collusion-resistant codes have been developed. The most popular type is the class of *bias-based* codes, introduced by G. Tardos in 2003. The initial paper [25], [26] was followed by improved analyses [3], [10], [11], [17], [23], [31], [30], code modifications [12], [20], [21], more advanced decoders [1], [5], [19], [22], [9], [27] and various generalizations [4], [28], [29], [32]. Bias-based codes have the advantage that they can achieve the asymptotically optimal relationship $\ell \propto c^2$ between the sufficient code length ℓ and the coalition size c.

One of the main advances in recent years was finding the *saddlepoint* of the information-theoretic max-min game [12], [15] in the case of joint decoding. Knowing the location of the saddlepoint allows the tracer to build a *universal* decoder that works optimally against the worst-case attack and that works well against all other attacks too.

II. Contributions and outline

We generalize the tally-based score function recently introduced by Škorić [27]. We combine a number (s) of neighbouring symbols in a user's codeword into a single composite symbol. If the original alphabet is denoted as $\mathcal{Q}$, the composite symbols are elements of $\mathcal{Q}^s$. We apply the tally-based score system of [27] to the composite symbols. The result is a Neyman-Pearson score that is based on more information than the original tally-based score.

The outline of this paper is as follows. In Section III we introduce notation and briefly review Tardos codes and the tally-based score function. In Section IV we derive our main result: a recipe for computing a universal score function for general s. Unfortunately the generic result is rather cumbersome; therefore we provide explicit formulas only up to $s = 4$ (Section V). We show that in the limit $c \to \infty$ the composite-symbol score reduces to a sum of $s = 1$ scores.

Section VI discusses the performance of our new score functions up to $s = 3$. ROC curves show a clear improvement when the collusion attack is Interleaving or Majority Voting. However, for the Minority Voting attack the performance is worse than [27].

In Section VII we apply our composite-symbol technique to the field of group testing. We derive a simple recipe for computing group-testing score functions for generic s.

III. Preliminaries

A. General notation and terminology

Random variables are written as capitals, and their realisations in lower-case. Sets are written in calligraphic font. (E.g. random variable $X \in \mathcal{X}$ with realisations x.) The probability of an event A is denoted as $\Pr[A]$, and the expectation over a random variable X is denoted as $\mathbb{E}_X[f(X)] \stackrel{\text{def}}{=} \sum_{x\in\mathcal{X}} \Pr[X = x]f(x)$. We define $[s] = \{1, \ldots, s\}$. Vectors are written in boldface. The 1-norm of a vector $\boldsymbol{v}$ is denoted as $|\boldsymbol{v}| = \sum_\alpha v_\alpha$. Falling factorials are written as $x^{(k)} \stackrel{\text{def}}{=} x(x-1)\cdots(x-k+1)$. The number of users is n. The alphabet is $\mathcal{Q}$, with size $|\mathcal{Q}| = q$. The length of the code is ℓ. Abstractly speaking, the content contains *positions* $i \in \{1, \ldots, \ell\}$; in each position a symbol from $\mathcal{Q}$ is embedded. The number of colluders is c. The set of colluders is denoted as $\mathcal{C} \subset [n]$ with $|\mathcal{C}| = c$. The coalition size that the code is built to withstand is c_0. (We will not always strictly distinguish between c and c_0.) The term 'asymptotically' will be used in the meaning 'coalition size going to infinity'.

B. Bias-based fingerprinting codes

The bias vector in position i is denoted as $\boldsymbol{p}_i = (p_{i\alpha})_{\alpha\in\mathcal{Q}}$. It satisfies $|\boldsymbol{p}_i| \stackrel{\text{def}}{=} \sum_{\alpha\in\mathcal{Q}} p_{i\alpha} = 1$. The bias vectors $\boldsymbol{p}_i$ are independently drawn from a probability density F. The asymptotically optimal F is given by the following

Dirichlet distribution (multivariate Beta distribution): $F(\boldsymbol{p}) = \Gamma(\frac{q}{2})[\Gamma(\frac{1}{2})]^{-q}\prod_{\alpha\in\mathcal{Q}} p_\alpha^{-1/2}$. We use the 'bar' notation to indicate a quantity in all positions, e.g. $\bar{\boldsymbol{p}} \stackrel{\text{def}}{=} (\boldsymbol{p}_i)_{i\in[\ell]}$.

The code matrix is a matrix $x \in \mathcal{Q}^{n\times\ell}$; the matrix rows are the codewords. The j'th row is denoted as $\bar{x}_j \stackrel{\text{def}}{=} (x_{ji})_{i\in[\ell]}$. The entries of x are generated column-wise from the bias vectors: in position i, the probability distribution for user j's symbol is given by $\Pr[X_{ji} = \alpha | \boldsymbol{P}_i = \boldsymbol{p}_i] = p_{i\alpha}$.

For $i \in [\ell]$, $\alpha \in \mathcal{Q}$, tally variables are defined as follows,

$$\begin{aligned} t_{i\alpha} &\stackrel{\text{def}}{=} |\{j \in [n] :\ x_{ji} = \alpha\}| \\ m_{i\alpha} &\stackrel{\text{def}}{=} |\{j \in \mathcal{C} :\ x_{ji} = \alpha\}|. \end{aligned} \tag{1}$$

In words: $t_{i\alpha}$ is the number of users who have symbol α in the i'th position of their codeword; $m_{i\alpha}$ is the number of *colluders* who have symbol α in the i'th position of their codeword. We write $\boldsymbol{t}_i = (t_{i\alpha})_{\alpha\in\mathcal{Q}}$ and $\boldsymbol{m}_i = (m_{i\alpha})_{\alpha\in\mathcal{Q}}$. These tallies satisfy $|\boldsymbol{t}_i| = n$ and $|\boldsymbol{m}_i| = c$.

In the Restricted Digit Model (RDM), in each position the colluders are allowed to output only a single symbol $y \in \mathcal{Q}$ with nonzero tally m_y, which symbol then gets detected with 100% fidelity by the tracer. As is customary in the literature on traitor tracing, we will assume that the attackers equally share the risk. This leads to "colluder symmetry", i.e. the attack is invariant under permutation of the colluder identities. Furthermore we assume that there is no natural ordering on the alphabet $\mathcal{Q}$. Given these two symmetries, the attack depends only on $\bar{\boldsymbol{m}}$, the set of colluder tallies. Any attack strategy can then be fully characterized by a set of probabilities $\theta_{\bar{\boldsymbol{y}}|\bar{\boldsymbol{m}}}$.

The process of generating the matrix x is fully position-symmetric, i.e. invariant under permutations of the columns of x. However, that does not guarantee that the optimal collusion strategy is position-symmetric as well, since the realisation of x itself breaks the symmetry. Asymptotically the symmetry is restored (due to $\ell \to \infty$); the attack strategy can then be parametrized more compactly as a set of probabilities $\theta_{y|\boldsymbol{m}}$ applied in each position independently. In the RDM the asymptotically optimal attack [13], [15] is the Interleaving attack: a colluder is selected uniformly at random and his symbol is output.

The process of tracing colluders based on $\bar{\boldsymbol{p}}$, x and $\bar{y}$ is referred to as 'decoding'. The decoder outputs a list $\mathcal{L} \subset [n]$ of suspicious users. Often a thresholding procedure is used: a score is computed for each user, or tuple of users, and they whose score exceeds the threshold are accused. In this scenario a decoder can make two kinds of mistake: (i) Accusation of one or more innocent users, known as False Positive (FP); (ii) Not finding any of the colluders, known as False Negative (FN). The error probabilities of the decoder are $P_{\text{FP}} = \Pr[\mathcal{L} \setminus \mathcal{C} \neq \emptyset]$ and $P_{\text{FN}} = \Pr[\mathcal{L} \cap \mathcal{C} = \emptyset]$.

C. Tally-based score function in a single position

Practically all known score functions assign a score in each position $i \in [\ell]$ and then take a sum over all positions to obtain a user's/tuple's overall score. The tally-based decoder proposed in [27] also has this structure. It was derived from the Neyman-Pearson hypothesis test for the hypothesis '$j \in \mathcal{C}$' versus '$j \notin \mathcal{C}$' for a user $j \in [n]$, making use of the properties of hypergeometric-distributed random variables (see Section III-D). It has the form of a log-likelihood ratio. The score for user j is given by $\sum_{i\in[\ell]} h(x_{ji}, y_i, \boldsymbol{t}_i)$, with

$$h(x, y, \boldsymbol{t}) = \delta_{xy} \ln(1 + \frac{1}{c_0 - 1} \cdot \frac{n-1}{t_y - 1}). \tag{2}$$

Although the hypothesis concerns a single user, the score takes into account the symbols received by other users via the tallies $\bar{\boldsymbol{t}}$.

D. The multivariate hypergeometric distribution

Consider a single column of the matrix x. Let $\boldsymbol{T}$ be the total tally vector and $\boldsymbol{M}$ the colluders' tally vector, as defined in (1). If a coalition of c users is selected uniformly at random out of the n users, the probability $L_{\boldsymbol{m}|\boldsymbol{t}}$ that colluder tally $\boldsymbol{m}$ occurs, for given $\boldsymbol{t}$, is

$$L_{\boldsymbol{m}|\boldsymbol{t}} \stackrel{\text{def}}{=} \Pr[\boldsymbol{M} = \boldsymbol{m} | \boldsymbol{T} = \boldsymbol{t}] = \frac{1}{\binom{n}{c}} \prod_{\alpha\in\mathcal{Q}} \binom{t_\alpha}{m_\alpha}. \tag{3}$$

(For each symbol α, a number m_α of users have to be selected out of the t_α users who have that symbol). Eq. (3) is known as the multivariate hypergeometric distribution. Let $\boldsymbol{r} = (r_\alpha)_{\alpha\in\mathcal{Q}}$, with $r_\alpha \in \mathbb{N}$. Moments of the hypergeometric distribution can be obtained from

$$\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}} \prod_{\alpha\in\mathcal{Q}} M_\alpha^{(r_\alpha)} = \frac{c^{(|\boldsymbol{r}|)}}{n^{(|\boldsymbol{r}|)}} \prod_{\alpha\in\mathcal{Q}} t_\alpha^{(r_\alpha)}. \tag{4}$$

(See e.g. [2]). In particular, the first and second moment are given by

$$\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}} \boldsymbol{M} = \frac{c}{n}\boldsymbol{t}, \tag{5}$$

$$\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}} M_\alpha M_\beta = \frac{c^{(2)}}{n^{(2)}} t_\alpha t_\beta + \frac{c(n-c)}{n^{(2)}} \delta_{\alpha\beta} t_\alpha. \tag{6}$$

IV. Generalised tally-based score

A. Composite symbols

Instead of looking at user codewords symbol-by-symbol, we consider combinations of s consecutive locations. For simplicity we assume that $\ell = L \cdot s$, with $L \in \mathbb{N}$. For $a \in [L]$ we define a sequence of s symbols as follows,

$$\xi_{j,a} \stackrel{\text{def}}{=} (x_{j,(a-1)s+1}, x_{j,(a-1)s+2}, \ldots, x_{j,as}). \tag{7}$$

I.e. ξ_{j1} is the first s-sequence and ξ_{jL} is the last s-sequence in user j's codeword. We view ξ_{ja} as a *composite symbol* taking values in the alphabet $\mathcal{Q}^s$. In analogy with (7) we define sequences in y as $\lambda_a \stackrel{\text{def}}{=} (y_{(a-1)s+1}, \ldots, y_{as}) \in \mathcal{Q}^s$. For $\beta \in \mathcal{Q}^s$ we define the tally $t_{a\beta}$ as the number of users whose sequence ξ_{ja} exactly equals β.

Let $\xi \in \mathcal{Q}^s$ and $J \subseteq [s]$. Then $\xi[J] \in \mathcal{Q}^{|J|}$ denotes a length-$|J|$ sub-sequence of ξ obtained by selecting the components of ξ indicated by J. For $\beta \in \mathcal{Q}^{|J|}$ we define $t^J_{a\beta}$ as the number of users whose J-part of ξ_{ja} equals β. We define a vector $\boldsymbol{e}_\xi$ of length q^s as $(\boldsymbol{e}_\xi)_\alpha = \delta_{\xi\alpha}$.

B. Derivation of the score function for arbitrary s

In [27] it was found that the Neyman-Pearson score for the hypothesis $j \in \mathcal{C}$ vs. $j \notin \mathcal{C}$ can be expressed as

$$\ln \frac{\mathbb{E}_{\bar{\boldsymbol{M}}|x,j\in\mathcal{C}}\theta_{\bar{y}|\bar{\boldsymbol{M}}}}{\mathbb{E}_{\bar{\boldsymbol{M}}|x,j\notin\mathcal{C}}\theta_{\bar{y}|\bar{\boldsymbol{M}}}} = \ln \frac{\mathbb{E}_{\bar{\boldsymbol{M}}|x,j\in\mathcal{C}}\prod_{i\in[\ell]}\theta_{y_i|\boldsymbol{M}_i}}{\mathbb{E}_{\bar{\boldsymbol{M}}|x,j\notin\mathcal{C}}\prod_{i\in[\ell]}\theta_{y_i|\boldsymbol{M}_i}}. \quad (8)$$

Evaluating the expectations over $\bar{\boldsymbol{M}}$ for given x involves computing a sum over all possible candidate coalitions, i.e. all size-c subsets of $[n]$. When n is of order 10^5 or larger, this is infeasible already for moderate c. In order to get an expression that can be handled more easily, some of the information in x has to be 'forgotten'. In [27] the solution was to discard everything except the codeword $\bar{x}_j$ and the tallies $\bar{\boldsymbol{t}}$. This leads to a complete factorization into single-position scores.

Furthermore, in [27] the Interleaving strategy $\theta_{y_i|\boldsymbol{m}_i} = m_{iy_i}/c$ was substituted, resulting in (2). For $c \to \infty$ the Interleaving attack lies in the mutual information maxmin game saddlepoint [15], [14], and for finite c it lies very close to this point; substitution of the saddlepoint-θ into the Neyman-Pearson score results in a *universal* score function, i.e. a score that not only performs optimally against the saddlepoint-value of the attack but also performs well against all other attacks. Hence, substituting the Interleaving attack into (8) yields an almost universal score function.

We now follow the same approach with respect to θ, but we discard less information from x. We 'remember' $\bar{x}_j$ and the composite-symbol tallies $t_{a\zeta}$ for $a \in [\ell/s], \zeta \in \mathcal{Q}^s$. This results in a score that is a sum of s-sequence sub-scores.

Theorem 1: Let $\ell = s$. Let $\xi \in \mathcal{Q}^s$ be shorthand notation for the codeword $\bar{x}_j$ of the user under scrutiny. Let $\lambda \in \mathcal{Q}^s$ be the colluders' output. Then the Neyman-Pearson score (8) for user j is equivalent to

$$w(\xi,\lambda,\boldsymbol{t}) \stackrel{\text{def}}{=} \ln\Big(\frac{\binom{n}{c}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta_{\lambda|\boldsymbol{M}}}{\binom{n-1}{c}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}-\boldsymbol{e}_\xi}\theta_{\lambda|\boldsymbol{M}}} - 1\Big) \quad (9)$$

where $\boldsymbol{M}$ and $\boldsymbol{t}$ are defined over $\mathcal{Q}^s$.

The proof is given in Appendix A.

Eq. (9) can be rewritten in many different ways. The presented form has the advantage that, once an analytic expression has (laboriously) been found for the numerator, a formula for the denominator can simply be obtained by replacing $\boldsymbol{t} \to \boldsymbol{t} - \boldsymbol{e}_\xi$. Note that the fraction $\binom{n}{c}/\binom{n-1}{c}$ in (9) simplifies to $n/(n-c)$.

We now consider $\ell = Ls$, with $L > 1$, as explained in Section IV-A, and look again at the general score expression (8). On the one hand we will keep track of the composite symbols tallies *within* each bunch of s columns, but on the other hand we 'forget', except for user j, how these composite symbols are organised into codewords.[1] According to Theorem 1 this leads to the following score system,

$$r_j = \sum_{a=1}^{L} w(\xi_{ja},\lambda_a,\boldsymbol{t}_a). \quad (10)$$

[1]For example, $q = 2$, $s = 3, \ell = 9$ and a user $\neq j$ has codeword 000111001. We take into account that there is a contribution to $t_{1,000}$ from the first bunch of s symbols, a contribution to $t_{2,111}$ from the second bunch and to $t_{3,001}$ from the third. However, we will forget *who* contributed what to the tallies, and thus we do *not* remember that the composite symbols 000, 111 and 001 are connected to each other.

where r_j is the score of user j, and the function w is defined in (9). Our next task is to compute the expectation $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta_{\lambda|\boldsymbol{M}}$ for one bunch of columns.

Lemma 1: Let the attack strategy be Interleaving. Then

$$\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta^{\text{Int}}_{\lambda|\boldsymbol{M}} = c^{-s}\sum_{z_1,\ldots,z_s\in\mathcal{Q}^s}\Big(\prod_{i=1}^{s}\delta_{z_i[i],\lambda[i]}\Big)\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\prod_{i=1}^{s}M_{z_i}. \quad (11)$$

Proof: The colluders apply the Interleaving strategy independently in each position, which yields $\theta_{\lambda|\boldsymbol{m}} = \prod_{i=1}^{s}(m^{\{i\}}_{\lambda[i]}/c)$. Furthermore, the sub-component tally $m^{\{i\}}_\alpha$, for $\alpha \in \mathcal{Q}$, can be expressed as

$$m^{\{i\}}_\alpha = \sum_{z\in\mathcal{Q}^s} m_z\delta_{z[i],\alpha}. \quad (12)$$

We use (12) s times, i.e. for $i = 1 \ldots s$, substituting $\alpha = \lambda[i]$. ∎

The expectation $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\prod_{i=1}^{s}M_{z_i}$ in (11) can be computed using (4). This leads to rather complicated expressions, especially for large s, since (11) contains powers of tallies M_{z_i}, whereas (4) works with falling factorials.

The powers that occur in the product $\prod_{i=1}^{s}M_{z_i}$ depend on the structure of the 'collisions' between $z_1,\ldots,z_s$, i.e. whether some of the z_i symbols are equal to each other and if so, which ones. This information can be captured in the notion of *partitions*. A partition of the set $[s]$ into k parts is defined as a set $\zeta = \{\zeta_1,\ldots,\zeta_k\}$ with $\zeta_a \subseteq [s]$, $\zeta_a \neq \emptyset$, $\zeta_a \cap \zeta_b = \emptyset$ for $a \neq b$ and $\bigcup_a \zeta_a = [s]$. We denote the space of partitions of $[s]$ as $\mathcal{P}_{[s]}$.

Theorem 2: It holds that

$$\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta^{\text{Int}}_{\lambda|\boldsymbol{M}} = \frac{1}{c^s n^{(s)}}\sum_{\zeta\in\mathcal{P}_{[s]}}\Lambda_{nc}(\zeta)\prod_{a=1}^{|\zeta|}t^{\zeta_a}_{\lambda[\zeta_a]} \quad (13)$$

where the $\Lambda_{nc}(\zeta)$ are expressions that depend only on n, c and ζ.

A proof sketch is given in Appendix B.

Corollary 1: The Neyman-Pearson score against the Interleaving attack is given by

$$g_s(\xi,\lambda,\boldsymbol{t}) \stackrel{\text{def}}{=} \quad (14)$$
$$\ln\Big[\frac{n-s}{n-c}\cdot\frac{\sum_{\zeta\in\mathcal{P}_{[s]}}\Lambda_{nc}(\zeta)\prod_{a=1}^{|\zeta|}t^{\zeta_a}_{\lambda[\zeta_a]}}{\sum_{\zeta\in\mathcal{P}_{[s]}}\Lambda_{n-1,c}(\zeta)\prod_{a=1}^{|\zeta|}(t^{\zeta_a}_{\lambda[\zeta_a]}-\delta_{\xi[\zeta_a]\lambda[\zeta_a]})} - 1\Big].$$

Proof: Follows from substituting (13) into (9). ∎

In general the parameters $\Lambda_{nc}(\zeta)$ are complicated, especially for large s. For $s = 2$, $s = 3$ and $s = 4$ we will give explicit results in the coming sections. The Λ parameter for the 'easiest' partition is given below for general s.

Lemma 2: Let $\zeta = \{\{1\},\{2\},\ldots,\{s\}\}$. Then $\Lambda_{nc}(\zeta) = c^{(s)}$.

Proof: We consider (11) and use (4). For all terms in the $z_1,\ldots,z_s$ summation, the expectation $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\prod_i M_{z_i}$ contains exactly one term that contains s powers of t, namely $\frac{c^{(s)}}{n^{(s)}}t_{z_1}\cdots t_{z_s}$. All the other terms contain fewer powers of t. Finally, performing the summations $\sum_{z_i}$ with the constraint $\delta_{z_i[i],\lambda[i]}$ yields factors $t^{\{i\}}_{\lambda[i]}$. ∎

Note that for $s = 1$ the g_s is equivalent to the known single-position score function h.

$$\begin{aligned} g_1(\xi,\lambda,\boldsymbol{t}) &= \ln(\frac{n-1}{n-c}\cdot\frac{t_y}{t_y-\delta_{xy}}-1) \\ &= \ln\frac{c-1}{n-c}+\ln(1+\frac{n-1}{c-1}\cdot\frac{\delta_{xy}}{t_y-\delta_{xy}}) \\ &= \ln\frac{c-1}{n-c}+h(x,y,\boldsymbol{t}). \end{aligned} \tag{15}$$

The constant shift $\ln\frac{c-1}{n-c}$ does not depend on $x, y, \boldsymbol{t}$ and therefore does not affect the score system.

C. Computational effort for computing the scores

How much computational effort is involved in computing the user scores g? First of all, the tally $t^J_{\lambda[J]}$ has to be computed for each subset $J \subseteq [s]$. There are $2^s - 1$ of these subsets. Each tally can be computed with practically the same amount of effort. Start with the s n-component vectors $(\delta_{\xi_j[1]\lambda[1]})_{j\in[n]} \cdots (\delta_{\xi_j[s]\lambda[s]})_{j\in[n]}$ and compute the $t^{\{i\}}_{\lambda[i]}$ from them by summing over the users. Then create the $\binom{s}{2}$ vectors $(\delta_{\xi_j[ik]\lambda[ik]})_{j\in[n]}$ by componentwise multiplication of the vectors $(\delta_{\xi_j[i]\lambda[i]})_{j\in[n]}$ and $(\delta_{\xi_j[k]\lambda[k]})_{j\in[n]}$. Store these vectors. Compute the $t^{\{i,k\}}_{\lambda[ik]}$ from them by summing over j, etc. Given code length ℓ, the total effort of computing all these tallies scales as $2^s n\ell/s$.

Next the sum over all partitions $\zeta \in \mathcal{P}_{[s]}$ has to be taken. The number of partitions is given by the Bell number B_s. For large s one can approximate $B_s \approx (\frac{s}{e\ln s})^s$ [7]. The number of multiplications needed in the summation terms is proportional to s. Finally, the number of composite-symbol scores that has to be computed is $n\ell/s$.

In conclusion, for large s the total effort involved in the computation of all users scores scales as $n\ell(\frac{s}{\ln s})^s$, and the main effort lies in multiplying the tallies in each term of the ζ-summation.

V. Scores for small s

A. Score for $s = 2$

Theorem 3: For $s = 2$ the Neyman-Pearson score (9) in the case of the Interleaving attack is given by

$$g_2(\xi,\lambda,\boldsymbol{t}) = \ln[-1+\frac{n-2}{n-c}\cdot \frac{(c-1)t^{\{1\}}_{\lambda[1]}t^{\{2\}}_{\lambda[2]}+(n-c)t_\lambda}{(c-1)(t^{\{1\}}_{\lambda[1]}-\delta_{\xi[1]\lambda[1]})(t^{\{2\}}_{\lambda[2]}-\delta_{\xi[2]\lambda[2]})+(n-1-c)(t_\lambda-\delta_{\xi\lambda})}]. \tag{16}$$

Proof: The expectation $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}M_{z_1}M_{z_2}$ follows from (6). Substitution into (11) gives $c^2\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta^{\mathrm{Int}}_{\lambda|\boldsymbol{M}} = \frac{c^{(2)}}{n^{(2)}}t^{\{1\}}_{\lambda[1]}t^{\{2\}}_{\lambda[2]} + \frac{c(n-c)}{n^{(2)}}t_\lambda$. Substitution of this expression, and of its shifted version with $(\boldsymbol{t}\to\boldsymbol{t}-\boldsymbol{e}_\xi, n\to n-1)$, into (9) yields (16). ■

B. Score for $s = 3$

Lemma 3: Let α,β,γ be symbols in some alphabet. Let $\boldsymbol{m}$ and $\boldsymbol{t}$ be the colluder tally and all-user tally respectively for this alphabet. Then

$$\begin{aligned} \mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}M_\alpha M_\beta M_\gamma &= \frac{c^{(3)}}{n^{(3)}}t_\alpha t_\beta t_\gamma \\ &+ \frac{c^{(2)}(n-c)}{n^{(3)}}t_\alpha(\delta_{\alpha\beta}t_\gamma+\delta_{\alpha\gamma}t_\beta+\delta_{\beta\gamma}t_\gamma) \\ &+ \frac{c(n-c)(n-2c)}{n^{(3)}}\delta_{\alpha\beta}\delta_{\beta\gamma}t_\alpha. \end{aligned} \tag{17}$$

Proof: Follows from (4) after some diligent work. ■

Theorem 4: For $s = 3$ the Neyman-Pearson score (9) in the case of the Interleaving attack is given by

$$g_3(\xi,\lambda,\boldsymbol{t}) = \ln[-1+\frac{n-3}{n-c}\cdot\frac{A_3}{B_3}], \text{ with} \tag{18}$$

$$\begin{aligned} A_3 &= c^{(3)}t^{\{1\}}_{\lambda[1]}t^{\{2\}}_{\lambda[2]}t^{\{3\}}_{\lambda[3]} \\ &\quad +c^{(2)}(n-c)(t^{\{1,2\}}_{\lambda[12]}t^{\{3\}}_{\lambda[3]}+t^{\{1,3\}}_{\lambda[13]}t^{\{2\}}_{\lambda[2]}+t^{\{2,3\}}_{\lambda[23]}t^{\{1\}}_{\lambda[1]}) \\ &\quad +c(n-c)(n-2c)t_\lambda \end{aligned} \tag{19}$$

$$B_3 = A_3 \text{ with } \boldsymbol{t}\to\boldsymbol{t}-\boldsymbol{e}_\xi,\ \ n\to n-1 \tag{20}$$

Proof: Follows the same steps as the proof of Theorem 3, but now starting from Lemma 3. ■

C. Score for $s = 4$

Lemma 4: Let $\alpha,\beta,\gamma,\varepsilon$ be symbols in some alphabet. Let $\boldsymbol{m}$ and $\boldsymbol{t}$ be the colluder tally and all-user tally respectively for this alphabet. Then

$$\begin{aligned} &n^{(4)}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}[M_\alpha M_\beta M_\gamma M_\varepsilon] = \\ &c^{(4)}t_\alpha t_\beta t_\gamma t_\varepsilon \\ &+c^{(3)}(n-c)[\delta_{\alpha\beta}t_\alpha t_\gamma t_\varepsilon+\delta_{\alpha\gamma}t_\alpha t_\beta t_\varepsilon+\delta_{\alpha\varepsilon}t_\alpha t_\beta t_\gamma \\ &\qquad +\delta_{\beta\gamma}t_\alpha t_\beta t_\varepsilon+\delta_{\beta\varepsilon}t_\alpha t_\beta t_\gamma+\delta_{\gamma\varepsilon}t_\alpha t_\beta t_\gamma] \\ &+c^{(2)}(n-c)^{(2)}[\delta_{\alpha\beta}\delta_{\gamma\varepsilon}t_\alpha t_\gamma+\delta_{\alpha\gamma}\delta_{\beta\varepsilon}t_\alpha t_\beta+\delta_{\alpha\varepsilon}\delta_{\beta\gamma}t_\alpha t_\beta] \\ &+c^{(2)}(n-c)(n-2c+1)[\delta_{\alpha\beta}\delta_{\beta\gamma}t_\alpha t_\varepsilon+\delta_{\alpha\beta}\delta_{\beta\varepsilon}t_\alpha t_\gamma \\ &\qquad +\delta_{\alpha\gamma}\delta_{\gamma\varepsilon}t_\alpha t_\beta+\delta_{\beta\gamma}\delta_{\gamma\varepsilon}t_\alpha t_\beta] \\ &+c(n-c)[(n-2c)(n-3c)+(n-c)]\delta_{\alpha\beta}\delta_{\beta\gamma}\delta_{\gamma\varepsilon}t_\alpha. \end{aligned} \tag{21}$$

Proof: Follows from (4) after diligent labour. ■

Theorem 5: For $s = 4$ the Neyman-Pearson score (9) in the case of the Interleaving attack has the form (14), with

$$\begin{aligned} \Lambda_{nc}(\{\{1\},\{2\},\{3\},\{4\}\}) &= c^{(4)} \\ \Lambda_{nc}(\{\{12\},\{3\},\{4\}\}) &= c^{(3)}(n-c) \\ \Lambda_{nc}(\{\{12\},\{34\}\}) &= c^{(2)}(n-c)^{(2)} \\ \Lambda_{nc}(\{\{123\},\{4\}\}) &= c^{(2)}(n-c)(n-2c+1) \end{aligned}$$

$$\Lambda_{nc}(\{\{1234\}\}) = c(n-c)[(n-2c)(n-3c)+(n-c)]. \tag{22}$$

The other Λ-parameters are obtained by permuting the set $[s]$.

Proof: Follows the same steps as the proof of Theorem 3, but now starting from Lemma 4. ■

D. Large-c asymptotics

We now study the large-c asymptotics of the score function (14). We define $\nu = \mathbb{E}[n/(cT_Y)]$. For most attack strategies it holds that $\nu \ll 1$ asymptotically. The Minority Voting attack is an exception. We look at the case $\nu \ll 1$. In Sections V-A to V-C we notice that each 'disappearance' of a factor t is accompanied by a factor $\approx n/c$. Thus, the dominant term in the ζ-summations in (14) comes from the partition $\{\{1\},\ldots,\{s\}\}$. The other terms are of relative order ν or smaller. We have

$$\begin{aligned}
&g_s(\xi,\lambda,\boldsymbol{t}) = \\
&\ln(-1+\frac{n-s}{n-c}\prod_{i=1}^{s}[1+\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}][1+\mathcal{O}(\frac{\nu}{n})]) \\
&= \ln(-1+\frac{n-s}{n-c}[1+\sum_{i=1}^{s}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}+\mathcal{O}(\frac{\nu}{n})]) \\
&= \ln\frac{c-s}{n-c} \\
&+\ln(1+\frac{n-s}{c-s}\sum_{i=1}^{s}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}+\mathcal{O}(\frac{\nu}{c})) \\
&= \ln\frac{c-s}{n-c} \\
&+\ln(1+\frac{n-1}{c-1}\sum_{i=1}^{s}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}+\mathcal{O}(\frac{\nu}{c})). \qquad (23)
\end{aligned}$$

On the other hand, the sum of single-position scores $\sum_{i=1}^{s} h(\xi[i],\lambda[i],\boldsymbol{t}^{\{i\}})$ can be written as

$$\begin{aligned}
&\sum_{i=1}^{s}\ln(1+\frac{n-1}{c-1}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}) \\
&= \ln\prod_{i=1}^{s}(1+\frac{n-1}{c-1}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}) \\
&= \ln(1+\frac{n-1}{c-1}\sum_{i=1}^{s}\frac{\delta_{\xi[i]\lambda[i]}}{t^{\{i\}}_{\lambda[i]}-\delta_{\xi[i]\lambda[i]}}+\mathcal{O}(\nu^2)). \qquad (24)
\end{aligned}$$

We see that asymptotically the composite-symbol score becomes equivalent to the sum of single-position scores, up to an unimportant term $\ln\frac{c-s}{n-c}$ which depends on neither the code matrix x nor the collusion output $\bar{y}$. This result is not surprising; the $s=1$ score is known to reach asymptotic capacity.

VI. Numerics

Fig. 1 shows simulation results. For each shown combination of parameters we have run the following experiment 10^6 times: generate $\bar{\boldsymbol{p}}$; generate the colluder symbols; apply the attack strategy to obtain $\bar{y}$; generate the δ_{xy}-matrix for the innocent users; compute the $t^J_{\lambda[J]}$ tallies; compute scores using $c_0 = c$. The ROC curve for each score function is plotted by varying the threshold Z. An estimate for P_{FP} is computed as the fraction of simulation runs in which it occurs that innocent scores exist above Z. Similarly, P_{FN} is estimated as the fraction of simulation runs in which all colluder scores lie below Z. Due to the finite number of runs, probabilities of order $1/\#\mathrm{runs}$ and smaller cannot be estimated accurately, as is evident from the jumps at the bottom of the graphs.

Note that we generated the bias vectors *without* a cutoff on $\boldsymbol{p}$-space. In the graphs we do not show the (symmetrized) Tardos score and the score of Oosterwijk et al. since they generally perform worse than the Laarhoven score.

In the ROC curves we notice the following trends:

- In case of the Interleaving attack there is a clear improvement in the effectiveness of the tally-based score when we go from $s=1$ (the score introduced in [27]) to $s=2$. The step from $s=2$ to $s=3$ has far less effect. All improvements become smaller with increasing c, in accordance with Section V-D.
- When there is a mismatch between the anticipated attack (Interleaving) and the actual attack, the composite tally-based score function is not necessarily the best one. Furthermore, with increasing s the performance can get worse. In a sense the mismatch between the actual attack and the expected attack gets worse with increasing s. More information is being used in the hypothesis test, but in the wrong way.

We conclude that our composite tally-based scores improve on the state of the art, but should be used as part of a battery of different score functions rather than stand-alone. When the attack is (close to) Interleaving, the new scores ($s \geq 2$) have the lowest error rates; in case of different attacks another score function (e.g. $s=1$) will catch the attackers.

VII. Group Testing

A. Connection between traitor tracing and group testing

There is a well known link [24], [6], [18], [16] between on the one hand Traitor Tracing in the Restrictive Digit Model with the 'All-1' attack, and on the other hand (non-adaptive) Group Testing [8]. The Group Testing scenario is as follows. There is a population of n people, of which c are infected. Medical tests are expensive, and there is money to do only ℓ tests, with $\ell \ll n$. Furthermore the tests take a long time, so they are done non-adaptively, in parallel. An efficient way has to be devised to find out who is infected. Luckily it is possible to combine samples (e.g. blood samples) from multiple people and run a single test on the combination; if one or more of the individual samples come from an infected person, the medical test is positive.

The analogy with Traitor Tracing is straightforward. The user symbol $x_{ji} \in \{0,1\}$ indicates whether person j'th blood is included in the i'th test. The result of the i'th test is $y_i \in \{0,1\}$. The way the combined test works exactly matches the All-1 strategy: $\theta_{1|m_1}$ equals 1 if $m_1 \geq 1$ and 0 if $m_1 = 0$.

B. Score function for group testing

In order to derive scores for group testing, we start from (9) and substitute the All-1 attack. The results are much simpler than for the Interleaving attack.

Theorem 6: Let $t^J_{\mathbf{0}}$ denote the number of users who have $\xi[J] = 0\cdots0$. Let $t^{\emptyset}_{\mathbf{0}} = n$. Furthermore let $Z(\lambda) \subseteq [s]$ denote

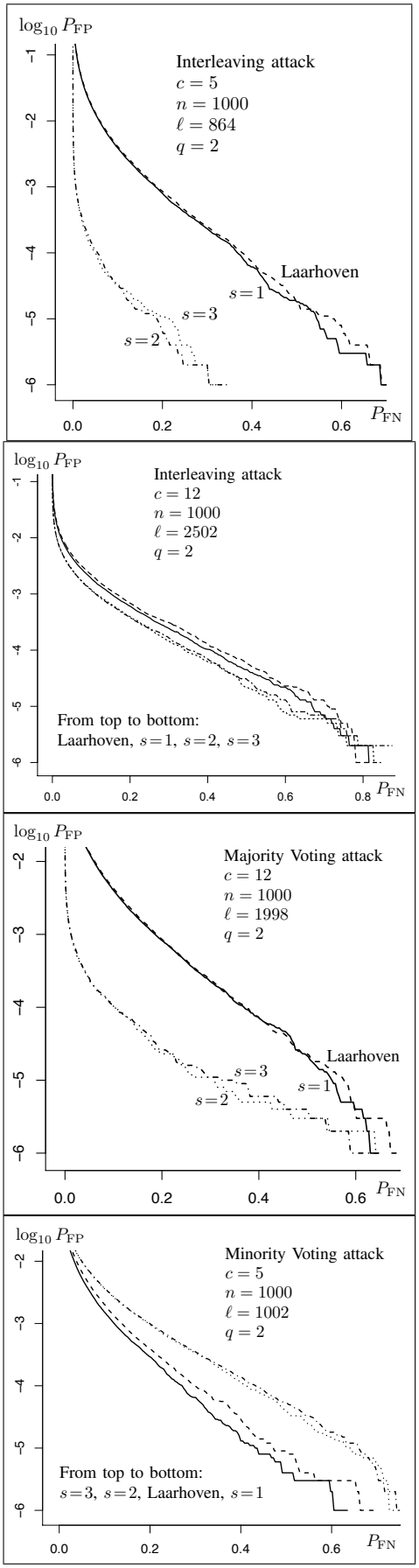

Fig. 1. *ROC curves for the Laarhoven score function, the tally-based score function ("$s = 1$") and the composite-symbol score functions with $s = 2$ and $s = 3$, for various attacks and parameter settings.*

the set of indices where $\lambda \in \mathcal{Q}^s$ contains a '0'.[2] Then

$$\binom{n}{c}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta^{\mathrm{All1}}_{\lambda|\boldsymbol{M}} = \sum_{J\subseteq[s]:\ Z(\lambda)\subseteq J} \binom{t^J_{\mathbf{0}}}{c} (-1)^{|J|-|Z(\lambda)|}. \quad (25)$$

In case $Z(\lambda)$ is empty, the term $J = \emptyset$ is part of the summation.

Proof: We introduce the abbreviation $D_\lambda \stackrel{\mathrm{def}}{=} \binom{n}{c}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}\theta^{\mathrm{All1}}_{\lambda|\boldsymbol{M}}$. We use induction on s, starting at $s = 1$. For $s = 1$ we get $D_0 = \binom{n}{c}\Pr_{\boldsymbol{M}|\boldsymbol{t}}[\boldsymbol{M} = c\boldsymbol{e}_0] = \binom{t_0}{c}$, and $D_1 = \binom{n}{c}\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}[1 - \theta^{\mathrm{All1}}_{0|\boldsymbol{M}}] = \binom{n}{c} - \binom{t_0}{c}$. This is consistent with the right hand side of (25). Next we assume that (25) holds for some $s \geq 1$. We append a '0' to the (composite) symbol $\lambda \in \mathcal{Q}^s$, creating a composite symbol $\lambda 0 \in \mathcal{Q}^{s+1}$. The appended 0 does not 'do' anything, in the sense that it only imposes the simple constraint that all colluders receive a 0 in position $s + 1$. The result $D_{\lambda 0}$ is obtained from (25) simply by appending a '0' in the index of the tally $t^J_{\mathbf{0}}$,

$$\begin{aligned} D_{\lambda 0} &= \sum_{J\subseteq[s]:\ Z(\lambda)\subseteq J} \binom{t^{J\cup\{s+1\}}_{\mathbf{0}}}{c} (-1)^{|J|-|Z(\lambda)|} \\ &= \sum_{J\subseteq[s]:\ Z(\lambda)\subseteq J} \binom{t^{J\cup\{s+1\}}_{\mathbf{0}}}{c} (-1)^{|J\cup\{s+1\}|-|Z(\lambda 0)|} \\ &= \sum_{J'\subseteq[s+1]:\ Z(\lambda 0)\subseteq J'} \binom{t^{J'}_{\mathbf{0}}}{c} (-1)^{|J'|-|Z(\lambda 0)|}. \quad (26) \end{aligned}$$

Eq. (26) exactly matches (25). Furthermore, we can permute the positions in $\lambda 0$; this yields $D_{\mathrm{perm}(\lambda 0)}$ expressions that also satisfy (25). The only symbol in Q^{s+1} that we have not covered yet is $11\cdots 1$. For this one we use $\sum_\lambda \theta_{\lambda|\boldsymbol{M}} = 1$.

$$\begin{aligned} D_{1^{s+1}} &= \binom{n}{c} - \sum_{\lambda\in\mathcal{Q}^{s+1}:\ \lambda\neq 1^{s+1}} D_\lambda \\ &= \binom{n}{c} - \sum_{\substack{\lambda\in\mathcal{Q}^{s+1}\\ \lambda\neq 1^{s+1}}} \sum_{\substack{J\subseteq[s+1]\\ Z(\lambda)\subseteq J}} \binom{t^J_{\mathbf{0}}}{c}(-1)^{|J|-|Z(\lambda)|} \\ &= \binom{n}{c} - \sum_{\substack{J\subseteq[s+1]\\ J\neq\emptyset}} \binom{t^J_{\mathbf{0}}}{c}(-1)^{|J|} \sum_{\substack{\lambda\in\mathcal{Q}^{s+1}\\ \lambda\neq 1^{s+1}, Z(\lambda)\subseteq J}} (-1)^{-|Z(\lambda)|}. \end{aligned}$$

The λ-summation can be seen as a sum that starts with the unique λ satisfying $Z(\lambda) = J$, and then in increasingly more positions a 0 is flipped to 1. The number of choices how to flip b positions is $\binom{|J|}{b}$. Thus the λ-summation can be computed using the binomial sum rule as $\sum_{b=0}^{|J|-1}\binom{|J|}{b}(-1)^{|J|-b} = -1$. The result for $D_{1^{s+1}}$ is precisely of the form (25), with $\binom{n}{c}$ being the $J = \emptyset$ term. ∎

Substitution of (25) into (9) yields a full score system.

Example 1: For $s = 2$, Theorem 6 yields

$$\begin{aligned} D_{00} &= \binom{t_{00}}{c} \\ D_{01} &= \binom{t_{00}+t_{01}}{c} - \binom{t_{00}}{c} \quad (27) \\ D_{11} &= \binom{n}{c} - \binom{t_{00}+t_{01}}{c} - \binom{t_{00}+t_{10}}{c} + \binom{t_{00}}{c}. \end{aligned}$$

[2]For example, $Z(10110) = \{2, 5\}$; $Z(11) = \emptyset$.

The D_{10} follows by permuting the positions in the D_{01} result. Note that $t_{00}+t_{01}=t_{\mathbf{0}}^{\{1\}}$, $t_{00}+t_{10}=t_{\mathbf{0}}^{\{2\}}$, $t_{00}=t_{\mathbf{0}}^{\{1,2\}}$ and $n=t_{\mathbf{0}}^{\emptyset}$.

We briefly comment on the amount of work required to compute all user scores. First, all the tallies have to be computed. The effort is similar to the Interleaving attack case, of order $2^{s-|Z(\lambda)|}n\ell/s$.

The number of terms in (25) is $2^{s-|Z(\lambda)|}$, and each term requires a number of multiplications proportional to c. There are n users. The number of composite-symbol scores is ℓ/s. Hence the multiplication effort scales as $cn\ell 2^{s-|Z(\lambda)|}/s$.

VIII. Summary

We have introduced a new class of score functions for q-ary traitor tracing. It is obtained from the tally-based score function of [27] by combining s consecutive q-ary symbols from a user's codeword into a single composite symbol. For general s the score is given by the rather complicated expression (14). From our numerical experiments it seems that in practice one rarely needs more than $s=2$. When the attack is not Interleaving, other scores can perform better. Hence our new score functions should be used as part of a battery of different score functions.

We applied the composite-symbol technique to Group Testing; this yields the general result (25). Future work will show how much performance is gained. Note that the Neyman-Pearson approach to obtain score functions is particularly well suited here since the 'attack' θ is known precisely.

As other future work we mention: (i) Going to the Combined Digit Model [29] instead of the Restricted Digit Model. Eq. (9) is general enough to accommodate this. (ii) Study $q \geq 3$. (iii) Sorting the columns of the code matrix in order of increasing t_y, and apply a large s to the lowest-t_y columns. This binds the most informative columns (low t_y) together. (iv) Study the case $c > c_0$.

Acknowledgment

Thijs Laarhoven and Benne de Weger are thankfully acknowledged for useful discussions.

References

[1] E. Amiri and G. Tardos. High rate fingerprinting codes and the fingerprinting capacity. In *SODA 2009*, pages 336–345, 2009.

[2] Y.M.M. Bishop, S.E. Fienberg, and P.W. Holland. *Discrete multivariate analysis: theory and practice*. M.I.T. Press, 1975.

[3] O. Blayer and T. Tassa. Improved versions of Tardos' fingerprinting scheme. *Designs, Codes and Cryptography*, 48(1):79–103, 2008.

[4] A. Charpentier, C. Fontaine, T. Furon, and I.J. Cox. An asymmetric fingerprinting scheme based on Tardos codes. In *Information Hiding 2011*, volume 6958 of *LNCS*, pages 43–58. Springer, 2011.

[5] A. Charpentier, F. Xie, C. Fontaine, and T. Furon. Expectation maximization decoding of Tardos probabilistic fingerprinting code. In *SPIE Media Forensics and Security 2009*, page 72540, 2009.

[6] C.J. Colbourn, D. Horsley, and V.R. Syrotiuk. Frameproof codes and compressive sensing. In *48th Allerton Conference on Communication, Control, and Computing*, pages 985–990, 2010.

[7] N.G. de Bruijn. *Asymptotic methods in analysis (3rd ed.)*. Dover, 1981.

[8] R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

[9] T. Furon and M. Desoubeaux. Tardos codes for real. In *IEEE Workshop on Information Forensics and Security (WIFS) 2014*, 2014.

[10] T. Furon, A. Guyader, and F. Cérou. On the design and optimization of Tardos probabilistic fingerprinting codes. In *Information Hiding 2008*, volume 5284 of *LNCS*, pages 341–356. Springer, 2008.

[11] T. Furon, L. Pérez-Freire, A. Guyader, and F. Cérou. Estimating the minimal length of Tardos code. In *Information Hiding 2009*, volume 5806 of *LNCS*, pages 176–190, 2009.

[12] Y.-W. Huang and P. Moulin. Capacity-achieving fingerprint decoding. In *IEEE Workshop on Information Forensics and Security (WIFS) 2009*, pages 51–55, 2009.

[13] Y.-W. Huang and P. Moulin. On the saddle-point solution and the large-coalition asymptotics of fingerprinting games. *IEEE Transactions on Information Forensics and Security*, 7(1):160–175, 2012.

[14] Y.-W. Huang and P. Moulin. On the fingerprinting capacity games for arbitrary alphabets and their asymptotics. *IEEE Transactions on Information Forensics and Security*, 9(9):1477–1499, 2014.

[15] Ye.-W. Huang and P. Moulin. On fingerprinting capacity games for arbitrary alphabets and their asymptotics. In *IEEE International Symposium on Information Theory (ISIT) 2012*, pages 2571–2575, 2012.

[16] T. Laarhoven. Efficient probabilistic group testing based on traitor tracing. In *51st Allerton Conference on Communication, Control and Computing*, pages 1458–1465, 2013.

[17] T. Laarhoven and B. de Weger. Optimal symmetric Tardos traitor tracing schemes. *Designs, Codes and Cryptography*, pages 1–21, 2012.

[18] P. Meerwald and T. Furon. Group testing meets traitor tracing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2011*, pages 4204–4207, 2011.

[19] P. Meerwald and T. Furon. Towards Joint Tardos Decoding: The 'Don Quixote' Algorithm. In *Information Hiding 2011*, pages 28–42, 2011.

[20] K. Nuida. Short collusion-secure fingerprint codes against three pirates. In *Information Hiding 2010*, volume 6387 of *LNCS*, pages 86–102. Springer, 2010.

[21] 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*, 52(3):339–362, 2009.

[22] J.-J. Oosterwijk, B. Škorić, and J. Doumen. Optimal suspicion functions for Tardos traitor tracing schemes. In *ACM Information Hiding and Multimedia Security Workshop (IH&MMSec) 2013*, pages 19–28, 2013.

[23] A. Simone and B. Škorić. Accusation probabilities in Tardos codes: beyond the Gaussian approximation. *Designs, Codes and Cryptography*, 63(3):379–412, 2012.

[24] D.R. Stinson, T. van Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, 86(2):595–617, 2000.

[25] G. Tardos. Optimal probabilistic fingerprint codes. In *ACM Symposium on Theory of Computing (STOC) 2003*, pages 116–125, 2003.

[26] G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2):1–24, 2008.

[27] B. Škorić. Tally-based simple decoders for traitor tracing and group testing. *IEEE Transactions on Information Forensics and Security*, 10(6):1221–1223, 2015.

[28] B. Škorić, S. Katzenbeisser, and M.U. Celik. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. *Designs, Codes and Cryptography*, 46(2):137–166, 2008.

[29] B. Škorić, S. Katzenbeisser, H.G. Schaathun, and M.U. Celik. Tardos Fingerprinting Codes in the Combined Digit Model. *IEEE Transactions on Information Forensics and Security*, 6(3):906–919, 2011.

[30] B. Škorić and J.-J. Oosterwijk. Binary and q-ary Tardos codes, revisited. *Designs, Codes, and Cryptography*, July 2013.

[31] B. Škorić, T.U. Vladimirova, M.U. Celik, and J.C. Talstra. Tardos Fingerprinting is Better Than We Thought. *IEEE Transactions on Information Theory*, 54(8):3663–3676, 2008.

[32] F. Xie, T. Furon, and C. Fontaine. On-off keying modulation and Tardos fingerprinting. In *Multimedia & Security (MM&Sec) 2008*, pages 101–106. ACM, 2008.

Appendix

A. Proof of Theorem 1

The score (8) is invariant under permutation of all the users other than j. Hence all the relevant information present in x is contained in the codeword $\bar{x}_j = \xi \in \mathcal{Q}^s$ and the tally vector $\boldsymbol{t}$ (which is now defined over $\mathcal{Q}^s$). For $\ell = s$ the $\bar{\boldsymbol{M}}$ corresponds to the composite-symbol tally $\boldsymbol{M}$, and the

sequence $\bar{y}$ corresponds to $\lambda \in \mathcal{Q}^s$. The expectation $\mathbb{E}_{\bar{\boldsymbol{M}}|x}$ equals $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t},\xi}$. The score (8) now takes the form

$$\ln \frac{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \frac{1}{\binom{n-1}{c-1}} \binom{t_\xi - 1}{m_\xi - 1} \prod_{\alpha \in \mathcal{Q}^s \setminus \{\xi\}} \binom{t_\alpha}{m_\alpha}}{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \frac{1}{\binom{n-1}{c}} \binom{t_\xi - 1}{m_\xi} \prod_{\alpha \in \mathcal{Q}^s \setminus \{\xi\}} \binom{t_\alpha}{m_\alpha}}. \tag{28}$$

Here we have used that for $j \in \mathcal{C}$ we have to choose $c-1$ colluders from $n-1$ users, while symbol ξ is 'used up' once by a *colluder*. For $j \notin \mathcal{C}$ we have to choose c colluders from $n-1$ users, with ξ being used up by an *innocent user*, which does not affect m_ξ. Next we discard the factors $\binom{n-1}{c-1}$ and $\binom{n-1}{c}$, since they lead to a constant offset of the logarithm, which has no effect on the score system. Then we use $\binom{t_\xi - 1}{m_\xi - 1} = \binom{t_\xi}{m_\xi} - \binom{t_\xi - 1}{m_\xi}$, allowing us to simplify the score to

$$\begin{aligned} &\ln\Big(\frac{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \prod_{\alpha \in \mathcal{Q}^s} \binom{t_\alpha}{m_\alpha}}{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \prod_{\alpha \in \mathcal{Q}^s} \binom{t_\alpha - \delta_{\alpha\xi}}{m_\alpha}} - 1\Big) \\ &= \ln\Big(\frac{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \prod_{\alpha \in \mathcal{Q}^s} \binom{t_\alpha}{m_\alpha}}{\sum_{\boldsymbol{m}} \theta_{\lambda|\boldsymbol{m}} \prod_{\alpha \in \mathcal{Q}^s} \binom{(\boldsymbol{t} - \boldsymbol{e}_\xi)_\alpha}{m_\alpha}} - 1\Big). \end{aligned} \tag{29}$$

Finally we use the definition (3) for the conditional probability $L_{\boldsymbol{m}|\boldsymbol{t}}$, taking into account that the tally vector $\boldsymbol{t} - \boldsymbol{e}_\xi$ pertains to $n-1$ users.

B. Proof (sketch) of Theorem 2

We start from (11). We write $\prod_{i=1}^{s} M_{z_i}$ as $\prod_{\alpha \in \mathcal{Q}^s} M_\alpha^{r_\alpha}$, with $\sum_\alpha r_\alpha = s$. Every increase of a counter r_α requires a Kronecker Delta of the form $\delta_{z_i z_j}$. Next we express powers in terms of falling factorials using Stirling numbers of the 2nd kind, $M_\alpha^{r_\alpha} = \sum_{k_\alpha=0}^{r_\alpha} \left\{ {r_\alpha \atop k_\alpha} \right\} M_\alpha^{(k_\alpha)}$. Then we compute the expectation $\mathbb{E}_{\boldsymbol{M}|\boldsymbol{t}}$ using (4). If $r_\alpha = 1$ then the result contains one power of t_α. For larger r_α, the powers of t_α that occur in the result are $t_\alpha^{r_\alpha}, t_\alpha^{r_\alpha - 1}, \ldots, t_\alpha$. All of these contributions can be written as a product of t_{z_i} factors, where the $\delta_{z_i z_j}$ factors cause the powers ≥ 1. The constraints on the composite symbols z_i, as imposed by the Kronecker Deltas $\delta_{z_i[i],\lambda[i]}$ in (11), when summed over in combination with the t_{z_i} tallies and the $\delta_{z_i z_j}$ factors, yield expressions of the form $t^J_{\lambda[J]} \stackrel{\text{def}}{=} \sum_z t_z \delta_{z[J],\lambda[J]}$ as defined in Section IV-A. For every $i \in [s]$ there is a $\delta_{z_i[i],\lambda[i]}$ constraint; hence the whole set $[s]$ is covered and the distribution of the constraints over the available t_{z_i} factors corresponds to a partition of $[s]$.