Laarhoven *EURASIP Journal on Information Security* (2016) 2016:3
DOI 10.1186/s13635-015-0026-8

EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Asymptotics of fingerprinting and group testing: capacity-achieving log-likelihood decoders

Thijs Laarhoven

Abstract

We study the large-coalition asymptotics of fingerprinting and group testing and derive explicit decoders that provably achieve capacity for many of the considered models. We do this both for simple decoders (which are fast but commonly require larger code lengths) and for joint decoders (which may be slower but achieve the best code lengths). We further make the distinction between informed decoding, where the pirate strategy is exactly known, and uninformed decoding, and we design decoding schemes for both settings.

For fingerprinting, we show that if the pirate strategy is known, the Neyman-Pearson-based log-likelihood decoders provably achieve capacity, regardless of the strategy. The decoder built against the interleaving attack is further shown to be a universal decoder, able to deal with arbitrary attacks and achieving the uninformed capacity. This universal decoder is shown to be closely related to the Lagrange-optimized decoder of Oosterwijk et al. and the empirical mutual information decoder of Moulin. Joint decoders are also proposed, and we conjecture that these also achieve the corresponding joint capacities.

For group testing, the simple decoder for the classical model is shown to be more efficient than the one of Chan et al. and it provably achieves the simple group testing capacity. For generalizations of this model such as noisy group testing, the resulting simple decoders also achieve the corresponding simple capacities.

Keywords: Fingerprinting, Traitor tracing, Group testing, Log-likelihood decoding, Hypothesis testing

1 Introduction

1.1 Fingerprinting

To protect copyrighted content against unauthorized redistribution, distributors commonly embed watermarks or fingerprints in the content, uniquely linking copies to individual users. Note that by watermarks/fingerprints, which may have a different meaning in other contexts, here we refer to the code words from collusion-resistant codes that are embedded in the content. If the distributor finds an unauthorized copy of the content online, he can then extract the watermark from this copy and compare it to the database of watermarks, to determine which user was responsible.

To combat this solution, a group of c pirates may try to form a coalition and perform a collusion attack. By comparing their unique versions of the content, they will detect differences in their copies which must be part of the watermark. They can then try to create a mixed pirate copy, where the resulting watermark matches the watermark of different pirates in different segments of the content, making it hard for the distributor to find the responsible users. The goal of the distributor of the content is to assign the watermarks to the users in such a way that, even if many pirates collude, the pirate copy can still be traced back to the responsible users.

1.2 Group testing

A different area of research that has received considerable attention in the last few decades is group testing, introduced by Dorfman [1] in the 1940s. Suppose a large population contains a small number c of infected (or defective) items. To identify these items, it is possible to perform group tests: testing a subset of the population will lead to a positive test result if this subset contains at least

Correspondence: mail@thijs.com
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

one defective item, and a negative result otherwise. Since the time to run a single test may be very long, the subsets to test need to be chosen in advance, after which all group tests are performed simultaneously. Then, when the test results come back, the subset of defective items needs to be identified. The goal of the game is to identify these defectives using as few group tests as possible and with a probability of error as small as possible.

1.3 Model

The above problems of fingerprinting and group testing can be jointly modeled by the following two-person game between (in terms of fingerprinting) the distributor $\mathcal{D}$ and the adversary $\mathcal{C}$ (the set of colluders or the set of defectives). Throughout the paper, we will mostly use terminology from fingerprinting (i.e., users instead of items, colluders instead of defective items), unless we are specifically dealing with group testing results.

First, there is a universe $\mathcal{U}$ of n users, and the adversary is assigned a random subset of users $\mathcal{C} \subseteq \mathcal{U}$ of size $|\mathcal{C}| = c$. The subset $\mathcal{C}$ is unknown to the distributor, but we assume that the distributor does know the size c of $\mathcal{C}$; there are various ways in practice to estimate the collusion size, and if necessary, multiple decoders can be deployed for different values of c, out of which the most accurate results could be used as the estimate for c. The aim of the game for the distributor is ultimately to discover $\mathcal{C}$. The two-person game consists of three phases: (1) the distributor uses an *encoder* to generate a fingerprinting code, used for assigning versions to users; (2) the colluders employ a *collusion channel* to generate the pirate output from their given code words; and (3) the distributor uses a *decoder* to map the pirate output to a set $\mathcal{C}' \subseteq \mathcal{U}$.

1.3.1 Encoder

The distributor generates a fingerprinting code $\mathcal{X}$ of n binary code words of length ℓ.[1] The parameter ℓ is referred to as the code length, and the distributor would like ℓ to be as small as possible. For the eventual embedded watermark, we assume that for each segment of the content, there are two differently watermarked versions, so the watermark of user j is determined by the ℓ entries in the jth code word of $\mathcal{X}$.

A common restriction on the encoding process is to assume that $\mathcal{X}$ is created by first generating a bias vector $\vec{P} \in (0,1)^\ell$ (by choosing each entry P_i independently from a certain distribution f_P) and then generating code words $\vec{X}_j \in \mathcal{X}$ according to $\mathbb{P}(X_{j,i} = 1) = P_i$. This guarantees that watermarks of different users j are independent and that watermarks in different positions i are independent. Fingerprinting schemes that satisfy this assumption are sometimes called bias-based schemes, and the encoders in this paper (both for group testing and fingerprinting) are also assumed to belong to this category.

1.3.2 Collusion channel

After generating $\mathcal{X}$, the code words are used to select and embed watermarks in the content, and the content is sent out to all users. The colluders then get together, compare their copies, and use a certain collusion channel or pirate attack $\vec{\Theta}$ to determine the pirate output $\vec{Y} \in \{0,1\}^\ell$. If the pirate attack behaves symmetrically both in the colluders and in the positions i, then the collusion channel can be modeled by a vector $\vec{\theta} \in [0,1]^{c+1}$, consisting of entries $\theta_z = f_{Y_i|Z_i}(1|z) = \mathbb{P}(Y_i = 1|Z = z)$ (for $z = 0, \ldots, c$) indicating the probability of outputting a 1 when the pirates received z ones and $c - z$ zeroes. A further restriction on $\vec{\theta}$ in fingerprinting is the marking assumption, introduced by Boneh and Shaw [2], which says that $\theta_0 = 0$ and $\theta_c = 1$, i.e., if the pirates receive only zeros or ones, they have to output this symbol.

1.3.3 Decoder

Finally, after the pirate output has been generated and distributed, we assume the distributor intercepts it and applies a decoding algorithm to $\vec{Y}$, $\mathcal{X}$, and $\vec{P}$ to compute a set $\mathcal{C}' \subseteq \mathcal{U}$ of accused users. The distributor wins the game if $\mathcal{C}' = \mathcal{C}$ (catch-all scenario) or $\emptyset \neq \mathcal{C}' \subseteq \mathcal{C}$ (catch-one scenario) and loses if this is not the case.

1.3.4 Fingerprinting vs. group testing

While the above model is described in fingerprinting terminology, it also covers many common group testing models. The users then correspond to items, the colluders translate to defectives, the code $\mathcal{X}$ corresponds to the group testing matrix X (where $X_{j,i} = 1$ if item j is included in the ith test), and the pirate output corresponds to positive/negative test results. The collusion channel is exactly what separates group testing from fingerprinting: while in fingerprinting it is commonly assumed that this channel is not known or only weakly known to the distributor, in group testing, this channel is usually assumed known in advance. This means that there is no malicious adversary in group testing but only a randomization procedure that determines $\vec{Y}$. Note also that in (noisy) group testing, the Boneh-Shaw marking assumption may not always hold.

1.4 Related work

Work on the fingerprinting game described above started in the late 1990s, and lower bounds on the code length were established of the order $\ell \propto c \ln n$ [2], until in 2003, Tardos [3] proved a lower bound of the order $\ell \propto c^2 \ln n$ and described a scheme with $\ell = O(c^2 \ln n)$, showing this bound is tight. Since the leading constants of the upper and lower bounds did not match, later work on fingerprinting focused on finding the optimal leading constant.

Based on channel capacities, Amiri and Tardos [4] and Huang and Moulin [5] independently derived the optimal leading constant to be 2 (i.e., an asymptotic code length of $\ell \sim 2c^2 \ln n$) and many improvements to Tardos's original scheme were made [6–9] to reduce the leading constant from 100 to $\frac{1}{2}\pi^2 \approx 4.93$. Recently, it was shown that with Tardos' original "score function" one cannot achieve capacity [10], which lead to the study of different score functions. Based on a result of Abbe and Zheng [11], Meerwald and Furon [12] noted that a score function designed against the worst-case attack achieves capacity against arbitrary attacks. This also lead to a proposal for a capacity-achieving score function in [13], which achieves the lower bound on the leading constant of 2.

Most of the work on fingerprinting focused on the setting of arbitrary, unknown attacks, but some work was also done on the informed setting, where the decoder knows or tries to estimate the pirate strategy [13–17]. It is well known that for suboptimal pirate attacks, the required code length may be significantly smaller than $\ell \sim 2c^2 \ln n$, but explicit schemes provably achieving an optimal scaling in ℓ are not known.

Research on the group testing problem started much longer ago, and already in 1985, exact asymptotics on the code length for probabilistic schemes were derived as $\ell \sim c \log_2 n$ [18], whereas deterministic schemes require a code length of $\ell \propto c^2 \ln n$ [19, 20]. Later work focused on slight variations of the classical model such as noisy group testing, where a positive result may not always correspond to the presence of a defective item due to "noise" in the test output [21–24]. For noisy group testing, exact asymptotics on the capacities (with leading constants) are yet unknown, and so it is not known whether existing constructions are optimal.

1.5 Contributions and outline

In this paper, we show how to build schemes using log-likelihood decoders that provably satisfy given bounds on the error probabilities and have a code length with the optimal asymptotic scaling. We do this both for the informed setting (where $\vec{\theta}$ is known to the decoder) and the universal setting, where $\vec{\theta}$ is only known to satisfy the marking assumption. The results for the informed setting for fingerprinting are summarized in Table 1. Simple informed and universal decoders are discussed in Sections 2 and 3, and joint informed and universal decoders are discussed in Sections 4 and 5, respectively. In between, Section 3.4 discusses a completely different approach to obtain a universal simple decoder (based on the empirical mutual information decoder of Moulin [25]) and shows how in the end the result is again quite similar.

2 Simple informed decoding

In this section, we will discuss simple decoders with explicit scheme parameters (code lengths, accusation thresholds) that provably satisfy given bounds on the error probabilities. The asymptotics of the resulting code lengths further show that these schemes are capacity-achieving; asymptotically, the code lengths achieve the lower bounds that follow from the simple capacities, as derived in [26].

Table 1 An overview of the provable asymptotic code lengths of the informed fingerprinting decoders discussed in this paper

Fingerprinting attack	Simple decoding	Joint decoding
$\vec{\theta}_{\text{int}}$: interleaving attack	$\ell \sim 2c^2 \ln n$	$\ell \sim 2c^2 \ln n$
$\vec{\theta}_{\text{all1}}$: all-1 attack	$\ell \sim \frac{c \ln n}{(\ln 2)^2}$	$\ell \sim c \log_2 n$
$\vec{\theta}_{\text{maj}}$: majority voting	$\ell \sim \pi c \ln n$	$\ell \sim c \log_2 n$
$\vec{\theta}_{\text{min}}$: minority voting	$\ell \sim \frac{c \ln n}{(\ln 2)^2}$	$\ell \sim c \log_2 n$
$\vec{\theta}_{\text{coin}}$: coin-flip attack	$\ell \sim \frac{4c \ln n}{(\ln 2)^2}$	$\ell \sim c \log_{5/4} n$

We will follow the bias-based and score-based framework introduced by Tardos [3], which was later generalized to joint decoders by Moulin [25]. For simple decoding, this means that a user j receives a score S_j of the form

$$S_j = \sum_{i=1}^{\ell} S_{j,i} = \sum_{i=1}^{\ell} g(x_{j,i}, y_i, p_i)$$

and he is accused iff $S_j \geq \eta$ for some fixed threshold η. The function g is sometimes called the score function. Note that since g only depends on $\mathcal{X}$ through $\vec{X}_j$, any decoder that follows this framework is a simple decoder.

2.1 Simple log-likelihood decoders

Several different score functions g have been considered before [3, 9, 13, 16], but in this work, we will restrict our attention to log-likelihood scores, which are known to perform well and which turn out to be quite easy to analyze.

First, when building a decoder, we naturally want to be able to distinguish between two cases: user j is guilty or user j is not guilty. To do this, we assign scores to users based on the available data, and we try to obtain an optimal trade-off between the false positive error (accusing an innocent user) and the false negative error (not accusing a guilty user). This problem is well known in statistics as a hypothesis testing problem, where in this case we want to distinguish between the following two hypotheses H_0 and H_1:

$$\begin{aligned} H_0 &: \quad \text{user } j \text{ is guilty } (j \in \mathcal{C}), \\ H_1 &: \quad \text{user } j \text{ is innocent } (j \notin \mathcal{C}). \end{aligned}$$

The Neyman-Pearson lemma [27] tells us that the most powerful test to distinguish between H_0 and H_1 is to test whether the following likelihood ratio exceeds an appropriately chosen threshold η:

$$\Lambda(\vec{x},\vec{y},\vec{p}) = \frac{f_{\vec{X},\vec{Y}|\vec{P}}(\vec{x},\vec{y}|\vec{p},H_0)}{f_{\vec{X},\vec{Y}|\vec{P}}(\vec{x},\vec{y}|\vec{p},H_1)}.$$

Taking logarithms, and noting that different positions i are i.i.d., it is clear that testing whether a user's likelihood ratio exceeds η_1 is equivalent to testing whether his score S_j exceeds $\eta = \ln \eta_1$ for g defined by

$$g(x,y,p) = \ln\left(\frac{f_{X,Y|P}(x,y|p,H_0)}{f_{X,Y|P}(x,y|p,H_1)}\right). \quad (1)$$

Thus, the score function g from (1) corresponds to using a Neyman-Pearson score over the entire code word $\vec{X}_j$, and therefore, g is in a sense optimal for minimizing the false positive error for a fixed false negative error. Score functions of this form were previously considered in the context of fingerprinting in, e.g., [17, 28], but these papers did not show how to choose η and ℓ to provably satisfy certain bounds on the error probabilities.

2.2 Theoretical evaluation

Let us first see how we can choose ℓ and η such that we can prove that the false positive and false negative error probabilities are bounded from above by certain values ε_1 and ε_2. For the analysis below, we will make use of the following function M, which is closely related to the moment-generating function of scores in one position i for both innocent and guilty users. For fixed p, this function M is defined on $[0,1]$ by

$$M(t) = \sum_{x,y} f_{X,Y|P}(x,y|p,H_0)^t f_{X,Y|P}(x,y|p,H_1)^{1-t}. \quad (2)$$

By writing out the corresponding expectations and scores $S_{j,i}$, it can be seen that the function M satisfies $M(t) = \mathbb{E}\left(e^{tS_{j,i}}|p,H_1\right)$ and $M(t) = \mathbb{E}\left(e^{(t-1)S_{j,i}}|p,H_0\right)$.

Theorem 1. *Let p and $\vec{\theta}$ be fixed and known to the decoder. Let $\gamma = \ln(1/\varepsilon_2)/\ln(n/\varepsilon_1)$, and let the code length ℓ and threshold η be chosen as*

$$\ell = \frac{\sqrt{\gamma}(1+\sqrt{\gamma}-\gamma)}{-\ln M(1-\sqrt{\gamma})} \ln\left(\frac{n}{\varepsilon_1}\right), \quad (3)$$

$$\eta = (1-\gamma)\ln\left(\frac{n}{\varepsilon_1}\right). \quad (4)$$

Then, with probability at least $1-\varepsilon_1$, no innocent users are accused, and with probability at least $1-\varepsilon_2$ at least one, colluder is caught.

Proof. For innocent users j, we would like to prove that $\mathbb{P}(S_j > \eta|H_1) \leq \frac{\varepsilon_1}{n}$, where S_j is the user's total score over all segments. If this can be proved, then since innocent users have independent scores, it follows that with probability at least $\left(1-\frac{\varepsilon_1}{n}\right)^n \geq 1-\varepsilon_1$, no innocent users are accused. To get somewhat tight bounds, we start by applying the Markov inequality to $e^{\alpha S_j}$ for some $\alpha > 0$:

$$\mathbb{P}(S_j > \eta|H_1) = \mathbb{P}\left(e^{\alpha S_j} > e^{\alpha\eta}|H_1\right) \leq \frac{\mathbb{E}\left(e^{\alpha S_j}|H_1\right)}{e^{\alpha\eta}}$$
$$= \frac{\prod_{i=1}^{\ell}\mathbb{E}\left(e^{\alpha S_{j,i}}|H_1\right)}{e^{\alpha\eta}} = \frac{M(\alpha)^{\ell}}{e^{\alpha\eta}}.$$

Let us now first switch to guilty users. We will prove that for some guilty user j, $\mathbb{P}(S_j < \eta|H_0) \leq \varepsilon_2$. Again using Markov's inequality with some fixed constant $\beta > 0$, we get

$$\mathbb{P}(S_j < \eta|H_0) = \mathbb{P}\left(e^{-\beta S_j} > e^{-\beta\eta}|H_0\right) \leq \frac{\mathbb{E}\left(e^{-\beta S_j}|H_0\right)}{e^{-\beta\eta}}$$
$$= \frac{\prod_{i=1}^{\ell}\mathbb{E}\left(e^{-\beta S_{j,i}}|H_0\right)}{e^{-\beta\eta}} = \frac{M(1-\beta)^{\ell}}{e^{-\beta\eta}}.$$

For both guilty and innocent users, we can now obtain bounds by choosing appropriate values for α and β. Investigating the resulting expressions, it seems that good choices for α, β leading to sharp bounds are $\alpha = 1-\sqrt{\gamma}$ and $\beta = \sqrt{\gamma}$. Substituting these choices for α and β, and setting the bounds equal to the desired upper bounds $\frac{\varepsilon_1}{n}$ and ε_2, we get

$$\mathbb{P}(S_j > \eta|H_1) \leq \frac{M(1-\sqrt{\gamma})^{\ell}}{e^{-\sqrt{\gamma}\eta}} e^{-\eta} = \frac{\varepsilon_1}{n},$$
$$\mathbb{P}(S_j < \eta|H_0) \leq \frac{M(1-\sqrt{\gamma})^{\ell}}{e^{-\sqrt{\gamma}\eta}} = \varepsilon_2.$$

Combining these equations, we obtain the given expression for η, and solving for ℓ leads to the expression for ℓ in (3). □

Compared to previous papers analyzing provable bounds on the error probabilities [3, 6, 7, 9, 29], the proof of Theorem 1 is remarkably short and simple. Note however that the proof that a colluder is caught assumes that the attack used by the colluders is the same as the one the decoder is built against and that the actual value of ℓ is still somewhat mysterious due to the term $M(1-\sqrt{\gamma})$. In Section 2.4, we will show how to get some insight into this expression for ℓ.

2.3 Practical evaluation

Before going into details how the code lengths of Theorem 1 scale, note that Theorem 1 only shows that with high probability we provably catch *at least one* colluder with this decoder. Although this is commonly the best you can hope for when dealing with arbitrary attacks in fingerprinting,[2] if the attack is colluder-symmetric, it is actually possible to catch *all* colluders with high probability. So instead, we would like to be able to claim that with high probability, the set of accused users $\mathcal{C}'$ *equals* the set of colluders $\mathcal{C}$. Similar to the proof for innocent users, we could simply replace ε_2 by $\frac{\varepsilon_2}{c}$ and argue that the probability of finding all pirates is the product of their individual probabilities of getting caught, leading to a lower bound on the success probability of $\left(1 - \frac{\varepsilon_2}{c}\right)^c \geq 1 - \varepsilon_2$. This leads to the following heuristic estimate for the code length required to catch *all* pirates.

Conjecture 1. *Let γ in Theorem 1 be replaced by $\gamma' = \ln(c/\varepsilon_2)/\ln(n/\varepsilon_1)$. Then, with probability at least $1 - \varepsilon_1$ no innocent users are accused, and with probability at least $1 - \varepsilon_2$, all colluders are caught when the (colluder-symmetric) pirate strategy matches the one predicted by the decoder.*

The problem with this claim is that the pirate scores are related through $\vec{Y}$, so they are not independent. As a result, we cannot simply take the product of the individual probabilities $\left(1 - \frac{\varepsilon_2}{c}\right)$ to get a lower bound on the success probability of $1 - \varepsilon_2$. On the other hand, especially when the code length ℓ is large and ε_2 is small, we do not expect the event $\{S_1 > T\}$ to tell us much about the probability of, e.g., $\{S_2 > T\}$ occurring. One might thus expect that $\{S_2 > T\}$ does not become much less likely when $\{S_1 > T\}$ occurs. But since it is not so simple to prove a rigorous upper bound on the *catch-all* error probability without assuming independence, we leave this problem for future work.

2.4 Asymptotic code lengths

Let us now see how the code lengths ℓ from (3) scale in terms of c and n. In general, this expression is not so pretty, but if we focus on the regime of large n (and fixed ε_1 and ε_2), it turns out that the code length always has the optimal asymptotic scaling, regardless of p and $\vec{\theta}$.

Theorem 2. *For large n and fixed ε_1 and ε_2, the code length ℓ of Theorem 1 scales as*

$$\ell = \frac{\log_2 n}{I(X_1; Y|P = p)}\left[1 + O(\sqrt{\gamma})\right], \tag{5}$$

where $I(X_1; Y|P = p)$ is the mutual information between a pirate symbol X_1 and the pirate output Y. As a result, ℓ has the optimal asymptotic scaling.

Proof. First, note that if $n \to \infty$ and $\varepsilon_1, \varepsilon_2$ are fixed, then $\gamma \to 0$. Let us first study the behavior of $M(1 - \sqrt{\gamma})$ for small γ, by computing the first order Taylor expansion of $M(1 - \sqrt{\gamma})$ around $\gamma = 0$. For convenience, below we abbreviate $f_{X,Y|P}(x, y|p)$ by $f(x, y|p)$.

$$\begin{aligned} &M(1 - \sqrt{\gamma}) \\ &= \sum_{x,y} f(x, y|p, H_0) \exp\left(-\sqrt{\gamma} \ln \frac{f(x, y|p, H_0)}{f(x, y|p, H_1)}\right) \\ &= \sum_{x,y} f(x, y|p, H_0) \left(1 - \sqrt{\gamma} \ln \tfrac{f(x,y|p,H_0)}{f(x,y|p,H_1)} + O(\gamma)\right) \\ &= 1 - \sqrt{\gamma} \sum_{x,y} f(x, y|p, H_0) \ln \frac{f(x, y|p, H_0)}{f(x, y|p, H_1)} + O(\gamma). \end{aligned}$$

Here, the second equality follows from the fact that if $f(x, y|p, H_0) = 0$, then the factor $f(x, y|p, H_0)$ in front of the exponentiation would already cause this term to be 0, while if $f(x, y|p, H_0) > 0$, then also $f(x, y|p, H_1) > 0$ and thus their ratio is bounded and does not depend on γ. Now, recognizing the remaining summation as the mutual information (in natural units) between a colluder symbol X_1 and the pirate output Y, we finally obtain:

$$M(1 - \sqrt{\gamma}) = 1 - I(X_1; Y|P = p)\sqrt{\gamma} \ln 2 + O(\gamma).$$

Substituting this result in the original equation for ℓ, and noting that the factor ε_1 inside the logarithm is negligible for large n, we finally obtain the result of (5). □

Note that in the discussion above, we did not make any assumptions on p. In fact, both Theorems 1 and 2 hold for *arbitrary* values of p; the decoder always achieves the capacity associated to that value of p. As a result, if we optimize and fix p based on $\vec{\theta}$ (using results from [26, Section II]), we automatically end up with a decoder that provably achieves capacity for this attack.

2.5 Fingerprinting attacks

Let us now consider specific pirate attacks which are often considered in the fingerprinting literature and investigate the resulting code lengths. We will again consider the following five attacks of [26], which were also considered in, e.g., [5, 12–17, 30–33]. Recall that $\theta_z = f_{Y|Z}(1|z)$ is the probability of the pirates outputting a 1 when they received z ones.

- Interleaving attack: The coalition randomly selects a pirate and outputs his symbol. This corresponds to

$$(\vec{\theta}_{\text{int}})_z = \frac{z}{c}. \qquad (0 \leq z \leq c)$$

- All-1 attack: The pirates output a 1 whenever they can, i.e., whenever they have at least one 1. This translates to

$$(\vec{\theta}_{\text{all1}})_z = \begin{cases} 0 & \text{if } z = 0; \\ 1 & \text{if } z > 0. \end{cases}$$

- Majority voting: The colluders output the most common received symbol. This corresponds to

$$(\vec{\theta}_{\text{maj}})_z = \begin{cases} 0 & \text{if } z < \frac{c}{2}; \\ 1 & \text{if } z > \frac{c}{2}. \end{cases}$$

- Minority voting: The colluders output the least common received symbol. This corresponds to

$$(\vec{\theta}_{\text{min}})_z = \begin{cases} 0 & \text{if } z = 0 \text{ or } \frac{c}{2} < z < c; \\ 1 & \text{if } z = c \text{ or } 0 < z < \frac{c}{2}. \end{cases}$$

- Coin-flip attack: If the pirates receive both symbols, they flip a fair coin to decide which symbol to output:

$$(\vec{\theta}_{\text{coin}})_z = \begin{cases} 0 & \text{if } z = 0; \\ \frac{1}{2} & \text{if } 0 < z < c; \\ 1 & \text{if } z = c. \end{cases}$$

Using Theorem 1, we can now obtain exact, provable expressions for the code length ℓ in terms of $\vec{\theta}, p, c, n, \varepsilon_1, \varepsilon_2$. In general, these expressions are quite ugly, but performing a Taylor series expansion around $c = \infty$ for the optimal values of p from [26, Section II.A], we obtain the following expressions for ℓ. Note that $\ell\left(\vec{\theta}_{\text{min}}\right) \sim \ell\left(\vec{\theta}_{\text{all1}}\right)$.

$$\ell\left(\vec{\theta}_{\text{int}}\right) = 2c^2 \ln\left(\frac{n}{\varepsilon_1}\right)\left[\frac{1+\sqrt{\gamma}-\gamma}{1-\sqrt{\gamma}} + O\left(\frac{1}{c^2}\right)\right], \tag{6}$$

$$\ell\left(\vec{\theta}_{\text{all1}}\right) = \frac{c\ln(n/\varepsilon_1)}{\ln(2)^2}\left[\tfrac{\sqrt{\gamma}(1+\sqrt{\gamma}-\gamma)\ln 2}{1-2^{-\sqrt{\gamma}}} + O\left(\tfrac{1}{c}\right)\right],$$

$$\ell\left(\vec{\theta}_{\text{maj}}\right) = \pi c \ln\left(\frac{n}{\varepsilon_1}\right)\left[\frac{1+\sqrt{\gamma}-\gamma}{1-\sqrt{\gamma}} + O\left(\frac{1}{c}\right)\right],$$

$$\ell\left(\vec{\theta}_{\text{coin}}\right) = \frac{4c}{\ln(2)^2}\ln\left(\frac{n}{\varepsilon_1}\right) \times \left[\frac{\sqrt{\gamma}(1+\sqrt{\gamma}-\gamma)\ln 2}{2-\left(1+\frac{1}{\sqrt{2}}\right)^{\sqrt{\gamma}}-\left(1-\frac{1}{\sqrt{2}}\right)^{\sqrt{\gamma}}} + O\left(\frac{1}{c}\right)\right].$$

If we assume that both $c \to \infty$ and $\gamma \to 0$, then we can further simplify the above expressions for the code lengths. The first terms between brackets all scale as $1 + O(\sqrt{\gamma})$, so the code lengths scale as the terms before the square brackets. These code lengths match the capacities of [26].

As for the score functions g, let us highlight one attack in particular, the interleaving attack. The all-1 decoder will be discussed in Section 2.6, while the score functions for other attacks can be computed in a similar fashion. For the interleaving attack, working out the probabilities in (1), we obtain the following score function:

$$g(x,y,p) = \begin{cases} \ln\left(1 + \frac{p}{c(1-p)}\right) & \text{if } x = y = 0; \\ \ln\left(1 - \frac{1}{c}\right) & \text{if } x \neq y; \\ \ln\left(1 + \frac{1-p}{cp}\right) & \text{if } x = y = 1. \end{cases}$$

2.6 Group testing models

For group testing, we will consider three models: the classical (noiseless) model and the models with additive noise and dilution noise. Other models where the probability of a positive test result only depends on the tally Z (such as the threshold group testing models considered in [26]) may be analyzed in a similar fashion. Note that the classical model is equivalent to the all-1 attack in fingerprinting, as was previously noted in, e.g., [17, 23, 34].

- Classical model: The test output is positive iff the tested pool contains at least one defective:

$$(\vec{\theta}_{\text{all1}})_z = \begin{cases} 0 & \text{if } z = 0; \\ 1 & \text{if } z > 0. \end{cases}$$

- Additive noise model: Just like the classical model, but if no defectives are tested, the result may still be positive:

$$(\vec{\theta}_{\text{add}})_z = \begin{cases} r & \text{if } z = 0; \\ 1 & \text{if } z > 0. \end{cases} \qquad (r \in (0,1))$$

- Dilution noise model: Similar to the classical model, but the probability of a positive result increases with z:

$$(\vec{\theta}_{\text{dil}})_z = \begin{cases} 0 & \text{if } z = 0; \\ 1 - r^z & \text{if } z > 0. \end{cases} \qquad (r \in (0,1))$$

We can again expand the expressions of Theorem 1 around $c = \infty$ for the optimal values of p from [26, Section II.B] but with the added parameter r, the resulting formulas are quite a mess. If we also let $\gamma \to 0$, then we can use Theorem 2 to obtain the following simpler expressions:

$$\ell\left(\vec{\theta}_{\text{all1}}\right) = \frac{c\ln n}{\ln(2)^2}\left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right],$$

$$\ell\left(\vec{\theta}_{\text{add}}\right) = \frac{c\ln n}{\ln(2)^2 - r\ln 2 + O(r^2)}\left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right],$$

$$\ell\left(\vec{\theta}_{\text{dil}}\right) = \frac{c\ln n}{\ln(2)^2 - O(r\ln r)}\left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right].$$

For more detailed expressions for ℓ, one may combine Theorems 1 and 2 with [26, Section II.B]. For the classical model, working out the details, we obtain the following result.

Corollary 1. *For the classical group testing model, the simple decoder for the optimal value $p = p^s_{all1} \approx \frac{\ln 2}{c}$ is given by*[3]

$$g\left(x,y,p^s_{all1}\right) = \begin{cases} +1 & \textit{if } (x,y) = (0,0); \\ -1 + O\left(\frac{1}{c}\right) & \textit{if } (x,y) = (0,1); \\ -\infty & \textit{if } (x,y) = (1,0); \\ +c & \textit{if } (x,y) = (1,1). \end{cases}$$

Using this decoder in combination with the parameters η and ℓ of Theorem 1, we obtain a simple group testing algorithm with an optimal asymptotic number of group tests of

$$\ell \sim \frac{c \ln n}{\ln(2)^2} \approx 2.08 c \ln n \approx 1.44 c \log_2 n.$$

This asymptotically improves upon results of, e.g., Chan et al. [35, 36] who proposed an algorithm with an asymptotic code length of $\ell \sim ec \ln n \approx 2.72 c \ln n$. Their algorithm does have a guarantee of never falsely identifying a non-defective item as defective (whereas our proposed decoder does not have this guarantee), but the price they pay is a higher asymptotic number of tests to find the defectives.

3 Simple universal decoding

While in the previous section we discussed simple decoders for the setting where $\vec{\theta}$ is completely known to the decoder, let us now consider the setting which is more common in fingerprinting, where the attack strategy is not assumed known to the decoder. This setting may partially apply to group testing as well (where there may be some unpredictable noise on the test outputs), but the main focus of this section is the uninformed fingerprinting game.

3.1 The simple interleaving decoder, revisited

Let us now try to investigate how to build a decoder that works against arbitrary attacks in fingerprinting. To build such a decoder, Meerwald and Furon [12] previously noted that Abbe and Zheng [11] proved in a more general context that under certain conditions on the set of allowed pirate strategies $\vec{\Theta}$, a decoder that works against the worst-case attack $\vec{\theta}^* \in \vec{\Theta}$ also works against arbitrary other attacks $\vec{\theta} \in \vec{\Theta}$. In this case, the set of allowed pirate strategies we consider is the set of all attacks satisfying the marking assumption $\mathcal{P}_{\text{mark}}$:

$$\mathcal{P}_{\text{mark}} = \left\{ \vec{\theta} \in [0,1]^{c+1} \mid \theta_0 = 0, \theta_c = 1 \right\}.$$

For finite c, the worst-case attack in fingerprinting (from an information-theoretic perspective) has been studied in, e.g., [5, 12], but in general, this attack is quite messy and unstructured. Since this attack is not so easy to analyze, let us therefore focus on the asymptotics of large c and n. Huang and Moulin [5] previously proved that for large coalitions, the optimal pirate attack is the interleaving attack. So combining this knowledge with the result of Abbe and Zheng, perhaps a good choice for a universal decoder is the interleaving decoder, which we recall is given by:

$$g(x,y,p) = \begin{cases} \ln\left(1 + \frac{p}{c(1-p)}\right) & \text{if } x = y = 0; \\ \ln\left(1 - \frac{1}{c}\right) & \text{if } x \neq y; \\ \ln\left(1 + \frac{1-p}{cp}\right) & \text{if } x = y = 1. \end{cases} \tag{7}$$

Let us take a closer look at this decoder. For fixed $\delta > 0$, if we look at values $p \in [\delta, 1-\delta]$ and focus on the regime of large c, we can perform a Taylor series expansion around $c = \infty$ to get $\ln(1+x) \sim x$. The resulting expressions then turn out to be closely related to Oosterwijk et al.'s [13] decoder h:

$$g(x,y,p) \sim \frac{h(x,y,p)}{c} = \begin{cases} +\frac{p}{c(1-p)} & \text{if } x = y = 0; \\ -\frac{1}{c} & \text{if } x \neq y; \\ +\frac{1-p}{cp} & \text{if } x = y = 1. \end{cases}$$

This implies that g and h are asymptotically equivalent for p sufficiently far away from 0 and 1. Since for Oosterwijk et al.'s score function one generally uses *cut-offs* on f_P (i.e., only using values $p \in [\delta, 1-\delta]$ for fixed $\delta > 0$) to guarantee that $h(x,y,p) = o(c)$ (cf. [29]), and since the decoder of Oosterwijk et al. is known to achieve capacity using these cut-offs, we immediately get the following result.

Proposition 1. *The score function g of* (7) *together with the bias density function (encoder) $f_P^{(\delta)}$ on $[\delta, 1-\delta]$ of the form*

$$f_P^{(\delta)}(p) = \frac{1}{(\pi - 4\arcsin\sqrt{\delta})\sqrt{p(1-p)}} \tag{8}$$

asymptotically achieve the simple capacity for the uninformed fingerprinting game when the same cut-offs δ as those of [29] are used.

So combining the log-likelihood decoder designed against the asymptotic worst-case attack (the interleaving attack) with the arcsine distribution with cut-offs, we obtain a universal decoder that works against arbitrary attacks.

3.2 Cutting off the cut-offs

Although Proposition 1 is already a nice result, the cut-offs δ have been a nagging inconvenience ever since Tardos introduced them in 2003 [3]. In previous settings, it was well known that this cut-off δ had to be large enough to guarantee that innocent users are not falsely accused and small enough to guarantee that large coalitions can still be caught. For instance, when using Tardos' original score function, it was impossible to do without cut-offs, and the same seems to hold for Oosterwijk et al.'s decoder h, since the scores blow up for $p \approx 0, 1$.

Looking at the universal log-likelihood decoder, one thing to notice is that the logarithm has a kind of mitigating effect on the tails of the score distributions. For

$0 \ll p \ll 1$, the resulting scores are roughly a factor c smaller than those obtained with h, but where the blow-up effect of h for small p is proportional to $\frac{1}{p}$, the function g only scales as $\ln\left(\frac{1}{p}\right)$ in the region of small p. This motivates the following claim, showing that with this decoder g, we finally do not need any cut-offs anymore!

Theorem 3. *The decoder* g *of* (7) *and the encoder* $f_P^*(p)$, *defined on* $[0,1]$ *by*

$$f_P^*(p) = \frac{1}{\pi\sqrt{p(1-p)}}, \tag{9}$$

together asymptotically achieve the simple capacity for the uninformed fingerprinting game.

Proof. We will argue that using this new universal decoder g, the difference in performance between using and not using cut-offs on f_P is negligible for large c. Since the encoder with cut-offs asymptotically achieves capacity, it then follows that without cut-offs, this scheme also achieves capacity.

Let us first prove that all moments of user scores are finite, even if no cut-offs are used. We will show that $\mathbb{E}[g(x,y,p)] < \infty$ for any x and y, so that after taking weighted combinations, we also get $\mathbb{E}(S_{j,i}|H_{0/1}) < \infty$. Let us consider the case where $x = y = 1$; other cases can be analyzed in the same way. Using the density function f_P^* of (9), we have

$$\begin{aligned} E &= \mathbb{E}[g(1,1,p)^k] \\ &= \int_0^1 \frac{dp}{\pi\sqrt{p(1-p)}} \log^k\left(1 + \frac{1-p}{cp}\right). \end{aligned}$$

Splitting the interval $[0,1]$ into two parts $[\delta,1]$ and $[0,\delta]$ (where δ depends on k but not on c), we obtain

$$\begin{aligned} E = &\int_\delta^1 \frac{dp}{\pi\sqrt{p(1-p)}} \log^k\left(1 + \frac{1-p}{cp}\right) \\ &+ \int_0^\delta \frac{dp}{\pi\sqrt{p(1-p)}} \log^k\left(1 + \frac{1-p}{cp}\right). \end{aligned}$$

Let us denote the two terms by E_1 and E_2. For the first term, we can perform a Taylor series expansion to obtain:

$$\begin{aligned} E_1 &= \int_\delta^1 \frac{dp}{\pi\sqrt{p(1-p)}} \log^k\left(1 + \frac{1-p}{cp}\right) \\ &= \int_\delta^1 \frac{dp}{\pi\sqrt{p(1-p)}} \left(\frac{1-p}{cp} + O\left(\frac{(1-p)^2}{c^2p^2}\right)\right)^k \\ &\leq \int_\delta^1 \frac{dp}{\pi\sqrt{p(1-p)}} \left(\frac{1}{c\delta} + O\left(\frac{1}{c^2\delta^2}\right)\right)^k \\ &\overset{(a)}{\leq} \int_\delta^1 \frac{dp}{\pi\sqrt{p(1-p)}} = 1 < \infty. \end{aligned}$$

Here, (a) follows from considering sufficiently large c while δ remains fixed. (Note that for large c, we even have $E_1 \to 0$.) For the other term, we do not expand the logarithm:

$$\begin{aligned} E_2 &= \int_0^\delta \frac{dp}{\pi\sqrt{p(1-p)}} \log^k\left(1 + \frac{1-p}{cp}\right) \\ &\propto \int_0^\delta \frac{dp}{\sqrt{p}} \log^k\left(\frac{1}{p}\right) \quad \xrightarrow{(b)} \quad 0. \end{aligned} \tag{10}$$

The last step (b) follows from the fact that the integration is done over an interval of width δ, while the integrand scales as $\frac{1}{\sqrt{p}}$ times some less important logarithmic terms. For arbitrary k, we can thus let $\delta = \delta(k) \to 0$ as a function of k to see that this is always bounded. Similar arguments can be used to show that for other values of x, y we also have $\mathbb{E}[g(x,y,p)^k] < 0$.

As a result, all innocent and guilty user score moments are finite, and so for large c from the Central Limit Theorem, it follows that the distributions of user scores will converge to Gaussians. If the scores of innocent and guilty users are indeed Gaussian for large c, then as discussed in, e.g., [9, 13], all that matters for assessing the performance of the scheme are the mean and variance of both curves. Similar to (10), the effects of small cut-offs on the distribution function f_P are negligible as both means and variances stay the same up to small order terms. So indeed, in both cases, the "performance indicator" [13] asymptotically stays the same, leading to equivalent code lengths. □

Note that the same result does *not* apply to the score function h of Oosterwijk et al. [13], for which the effects of values $p \approx 0, 1$ are not negligible. The main difference is that for small p, the score function h scales as $\frac{1}{p}$ (which explodes when p is really small), while the log-likelihood decoder g only scales as $\ln\left(\frac{1}{p}\right)$. Figure 1 illustrates the difference in the convergence of normalized innocent user scores to the standard normal distribution, when using the score functions g and h. These are experimental results for $c = 10$ and $\ell = 10,000$ based on 10,000 simulated scores for each curve, and for both score functions, we did not use any cut-offs. As we can see, using g the normalized scores $\tilde{S}_j = (S_j - \mathbb{E}S_j)/\sqrt{\operatorname{Var} S_j}$ are close to Gaussian, while using h the curves especially do not look very Gaussian for $\tilde{S}_j \gg 0$; in most cases, the distribution tails are much too large. For the minority voting attack, the resulting curve does not even seem close to a standard normal Gaussian distribution.

3.3 Designing the scheme

With the above result in mind, let us now briefly discuss how to actually build a universal scheme with the interleaving decoder g. From Theorem 3, it is clear that for generating biases, we should use the arcsine distribution f_P^*, and our decoder will be the interleaving decoder g

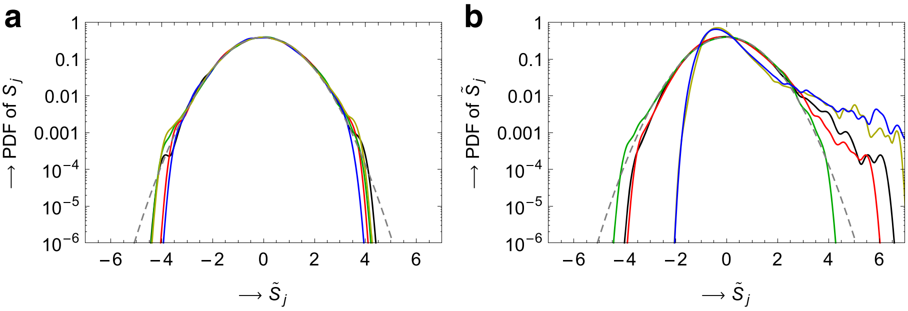

Fig. 1 Estimates of the probability density functions of normalized innocent user scores for $c = 10$ and $\ell = 10{,}000$ based on 10,000 simulations of innocent user scores for each attack. The different lines correspond to the interleaving attack (*black*), all-1 attack (*red*), majority voting (*green*), minority voting (*blue*), and the coin-flip attack (*yellow*), while the *gray dashed line* corresponds to the standard normal distribution with $f_Z(z) \propto \exp\left(-\frac{1}{2}z^2\right)$. Note that the drops near densities of 10^{-4} (the far ends of the curves) occur due to smoothening the curve with only 10^4 samples and do not reflect the real distribution tails

of (7). What remains is figuring out how to choose ℓ and η for arbitrary attacks.

First, it is important to note that the expected innocent and guilty scores per segment ($\mu_1 = \mathbb{E}(S_{j,i}|H_1)$ and $\mu_0 = \mathbb{E}(S_{j,i}|H_0)$), and the variance of the innocent and guilty scores ($\sigma_1^2 = \operatorname{Var}(S_{j,i}|H_1)$ and $\sigma_0^2 = \operatorname{Var}(S_{j,i}|H_0)$) heavily depend on the collusion channel $\vec{\theta}$. This was not the case for Tardos' original decoder [3] and the symmetrized decoder [9], for which η could be fixed in advance regardless of the collusion strategy. This means that we will either have to delay fixing η until the decoding stage, or scale/translate scores per segment accordingly at each position i.

For choosing the code length ℓ and threshold η, let us focus on the regime of reasonably large c. In that case, as argued above, the total innocent and guilty scores will behave like Gaussians, with parameters $S_1 \sim \mathcal{N}\left(\ell\mu_1, \ell\sigma_1^2\right)$ and $S_0 \sim \mathcal{N}\left(\ell\mu_0, \ell\sigma_0^2\right)$. To distinguish between these two distributions, using, e.g., Sanov's theorem the code rate, $\ell/\ln n$ should be proportional to the Kullbeck-Leibler divergence between the two distributions:

$$d(S_0\|S_1) = \frac{(\mu_0-\mu_1)^2}{\sigma_1^2} + \frac{1}{2}\left(\frac{\sigma_0^2}{\sigma_1^2} - 1 - \ln\frac{\sigma_0^2}{\sigma_1^2}\right).$$

A similar expression appears in [9, 13], where it was noted that $\sigma_0 \ll \sigma_1$, so that the first term is the most important term. In [13, 29], the ratio $\frac{(\mu_0-\mu_1)^2}{\sigma_1^2}$ was coined the "performance indicator," and it was argued that this ratio should be maximized. In [13], it was further shown that when using their decoder h, this ratio is minimized by the pirates when they choose the interleaving attack $\vec{\theta}_{\text{int}}$. In other words, assuming scores are Gaussian for large c, the best attack the pirates can use is the interleaving attack when using h as the decoder.

Since our new decoder g is very similar to Oosterwijk et al.'s decoder h (by $c \cdot g \approx h$), a natural conjecture would be that also for this new score function, asymptotically the best pirate attack maximizing the decoder's error probabilities is the interleaving attack. Experiments with g and previous experiments of [13] with h indeed show that other pirate attacks (such as those considered in Section 2.5) generally perform worse than the interleaving attack. As a result, a natural choice for selecting ℓ would be to base ℓ on the code length needed to deal with the (asymptotic) worst-case attack for this decoder, which we conjecture is the interleaving attack. And for the interleaving attack, we know how to choose ℓ by Theorem 1, Eq. 6, and Theorem 2:

$$\begin{aligned}\ell &= \frac{\sqrt{\gamma}(1+\sqrt{\gamma}-\gamma)}{-\ln M(1-\sqrt{\gamma})}\ln\left(\frac{n}{\varepsilon_1}\right)\\ &= 2c^2\ln\left(\frac{n}{\varepsilon_1}\right)\left[\frac{1+\sqrt{\gamma}-\gamma}{1-\sqrt{\gamma}} + O\left(\frac{1}{c^2}\right)\right]\\ &= 2c^2\ln n\left[1 + O\left(\sqrt{\gamma}+\frac{1}{c^2}\right)\right].\end{aligned}$$

These choices for ℓ thus seem reasonable estimates for the code lengths required to deal with arbitrary attacks.

Finally, for choosing η, as argued before, this parameter depends on the pirate strategy $\vec{\theta}$, which may lead to different scalings and translations of the curves of innocent and guilty user scores. What we could do is compute the parameters $\mu_{0/1}, \sigma_{0/1}^2$ after obtaining the pirate output $\vec{y}$ and normalize the scores accordingly. This means that after computing user scores S_j, we apply the following transformation:

$$\tilde{S}_j = \frac{S_j - \ell\mu_1}{\sqrt{\ell}\sigma_1}.$$

This guarantees that the scores of innocent users will roughly be distributed as a standard normal distribution, and for guilty users, this results in a normal distribution with mean $\sqrt{\ell}\left(\frac{\mu_0-\mu_1}{\sigma_1}\right)$ and variance $\frac{\sigma_0^2}{\sigma_1^2}$. To guarantee that the probability that an innocent user has a score below η is at least $1 - \frac{\varepsilon_1}{n}$, it suffices to let

$$\tilde{\eta} = \frac{\eta - \ell\mu_1}{\sqrt{\ell}\sigma_1} \approx \Phi^{-1}\left(1 - \frac{\varepsilon_1}{n}\right) \sim \sqrt{\ln\left(\frac{n}{\varepsilon_1}\right)},$$

where Φ denotes the distribution function of the standard normal distribution $\mathcal{N}(0, 1)$. This means that after transforming the scores, the threshold can be fixed independent of the pirate strategy.

3.4 Another simple universal decoder

Besides Oosterwijk et al.'s Lagrangian approach and our Neyman-Pearson-based approach to obtaining efficient decoders, let us now mention a third way to obtain a similar capacity-achieving universal decoder.

To construct this decoder, we start with the empirical mutual information decoder previously proposed by Moulin [25], and for now, let us assume $p_i \equiv p$ is fixed.[4] With this decoder, a user is assigned a score of the form

$$S_j = \sum_{x,y} \hat{f}_{X,Y|P}(x, y|p) \ln\left(\frac{\hat{f}_{X,Y|P}(x, y|p)}{\hat{f}_{X|P}(x|p)\hat{f}_{Y|P}(y|p)}\right)$$

and again the decision to accuse depends on whether this score exceeds some fixed threshold η. Here, $\hat{f}$ is the empirical estimate of the actual probability f, i.e., $\hat{f}_{X,Y|P}(x, y|p) = |\{i : (x_{j,i}, y_i) = (x, y)\}|/\ell$. Writing out the empirical probability outside the logarithm, and replacing the summation over x, y by a summation over the positions i, this is equivalent to

$$S_j = \frac{1}{\ell}\sum_{i=1}^{\ell} \ln\left(\frac{\hat{f}_{X,Y|P}(x_{j,i}, y_i|p)}{\hat{f}_{X,Y|P}(x_{j,i}, y_i|p, H_1)}\right).$$

Now, this almost fits the score-based simple decoder framework, except for that the terms inside the logarithm are not independent for different positions i. To overcome this problem, we could try to replace the empirical probabilities $\hat{f}$ by the actual probabilities f, but to compute $f_{X,Y|P}(x_{j,i}, y_i|p)$, we need to know whether user j is guilty or not. Solving this final problem using Bayesian inference, we get the following result.

Lemma 1. *Approximating the empirical probabilities in the empirical mutual information decoder with actual probabilities using Bayesian inference (with an a priori probability of guilt $\mathbb{P}(j \in \mathcal{C}) = \frac{c}{n}$), this decoder corresponds to using the following score function m:*

$$m(x, y, p) = \ln\left(1 + \frac{c}{n}\left[\frac{f_{X,Y|P}(x, y|p, H_0)}{f_{X,Y|P}(x, y|p, H_1)} - 1\right]\right). \tag{11}$$

Proof. The value of $f_{X,Y|P}(x_{j,i}, y_i|p, H_1)$ can be computed without any problems, so let us focus on the term $f_{X,Y|P}(x_{j,i}, y_i|p)$. Using Bayesian inference, we have

$$\begin{aligned} f_{X,Y|P}(x, y|p) = {} & \mathbb{P}(j \in \mathcal{C}) f_{X,Y|P}(x, y|p, H_0) \\ & + \mathbb{P}(j \notin \mathcal{C}) f_{X,Y|P}(x, y|p, H_1). \end{aligned}$$

Assuming an a priori probability of guilt of $\mathbb{P}(j \in \mathcal{C}) = \frac{c}{n}$ and dividing by $f_{X,Y|P}(x, y|p, H_1)$, we get

$$\frac{f_{X,Y|P}(x, y|p)}{f_{X,Y|P}(x, y|p, H_1)} = \frac{c}{n}\frac{f_{X,Y|P}(x, y|p, H_0)}{f_{X,Y|P}(x, y|p, H_1)} + 1 - \frac{c}{n}.$$

Taking logarithms, this leads to (11). □

Although this score function looks very similar to the log-likelihood decoder, there are some essential differences. For instance, for the all-1 attack, we have $g(1, 0, p) = -\infty$ while $m(1, 0, p) = \ln\left(1 - \frac{c}{n}\right) > -\infty$. For the interleaving attack, for which we may again hope to obtain a universal decoder using this approach, we do get a familiar result.

Corollary 2. *For the interleaving attack strategy, the Bayesian approximation of the empirical mutual information decoder of (11) satisfies*

$$m(x, y, p) = \begin{cases} \ln\left(1 + \frac{p}{n(1-p)}\right) & \textit{if } x = y = 0; \\ \ln\left(1 - \frac{1}{n}\right) & \textit{if } x \neq y; \\ \ln\left(1 + \frac{1-p}{np}\right) & \textit{if } x = y = 1. \end{cases} \tag{12}$$

For values of $p \in [\delta, 1 - \delta]$ with $\delta > 0$ and large c, this decoder is again equivalent to both the log-likelihood score function g and Oosterwijk et al.'s function h:

$$c \cdot g(x, y, p) \sim n \cdot m(x, y, p) \sim h(x, y, p).$$

For $p \approx 0, 1$, the logarithm again guarantees that scores do not blow up so much, but due to the factor n in the denominator (rather than a factor c, as in g), the scores relatively increase more when p approaches 0 than for the score function g.

4 Joint informed decoding

In this section, we will discuss informed joint decoders which we conjecture are able to find pirates with shorter code lengths than simple decoders. The asymptotics of the resulting code lengths further motivate that these schemes may be optimal but proving that they are indeed optimal remains an open problem.

Following the score-based framework for joint decoders of Moulin [25], we assign tuples T of size c a score of the form

$$S_T = \sum_{i=1}^{\ell} S_{T,i} = \sum_{i=1}^{\ell} g(\{x_{j,i} : j \in T\}, y_i, p_i).$$

Note that if $\vec{\theta}$ is colluder-symmetric, then this is equivalent to

$$S_T = \sum_{i=1}^{\ell} g(z_{T,i}, y_i, p_i),$$

where $z_{T,i} = \sum_{j \in T} x_{j,i}$ is the tally of the number of ones received by the tuple T in position i. For the accusation phase, we now accuse all users in T iff $S_T \geq \eta$ for some fixed threshold η. Note that this accusation algorithm is not exactly well-defined, since it is possible that a user appears both in a tuple that is accused and in a tuple that is not accused. For the analysis, we will assume that the scheme is only successful if the single tuple consisting of all colluders has a score exceeding η and no other tuples have a score exceeding η, in which case all users in that guilty tuple are accused. This may be too pessimistic for evaluating the performance of the scheme.

4.1 Joint log-likelihood decoders

For building a joint decoder, we would like to be able to distinguish between the all-guilty tuple and other tuples, when focusing on the catch-all scenario, so a natural generalization of the hypotheses H_0 and H_1 for simple decoding would be to let $H_0 : T = \mathcal{C}$ and $H_1 : T \neq \mathcal{C}$. However, with this choice of H_1, computing probabilities $f_{Z,Y|P}(z, y|p, H_1)$ is complicated: the event H_1 does not completely determine $f_{Y|Z}(y|z)$, since that depends on exactly how many colluders are present in T. To be able to compute the likelihood ratios, we therefore use the following two hypotheses, which were also used in, e.g., [12]:

$$\begin{aligned} H_0 &: \quad \text{all users } j \in T \text{ are guilty } (T = \mathcal{C}), \\ H_1 &: \quad \text{all users } j \in T \text{ are innocent } (T \cap \mathcal{C} = \varnothing). \end{aligned}$$

We again use the corresponding log-likelihood ratio per position as our score function g, which is again the logarithm of the likelihood ratio over all positions i:

$$g(z, y, p) = \ln \left(\frac{f_{Z,Y|P}(z, y|p, H_0)}{f_{Z,Y|P}(z, y|p, H_1)} \right). \tag{13}$$

Using this joint score function, g corresponds to a most powerful test according to the Neyman-Pearson lemma [27], so g is in a sense optimal for distinguishing between H_0 and H_1. Joint decoders of this form were previously considered in, e.g., [12].

4.2 Theoretical evaluation

Let us again study how to choose ℓ and η such that we can prove that the false positive and false negative error probabilities are bounded from above by certain values ε_1 and ε_2. Below we will again make use of the function M of (2) where the simple hypotheses have been replaced by our new joint hypotheses H_0 and H_1.

Theorem 4. *Let p and $\vec{\theta}$ be fixed and known to the decoder. Let $\gamma = \ln(1/\varepsilon_2)/\ln(n^c/\varepsilon_1)$, and let the code length ℓ and the threshold η be defined as*

$$\ell = \frac{\sqrt{\gamma}(1 + \sqrt{\gamma} - \gamma)}{-\ln M(1 - \sqrt{\gamma})} \ln \left(\frac{n^c}{\varepsilon_1} \right).$$

$$\eta = (1 - \gamma) \ln \left(\frac{n^c}{\varepsilon_1} \right),$$

Then, with probability at least $1 - \varepsilon_1$ all all-innocent tuples are not accused, and with probability at least $1 - \varepsilon_2$, the single all-guilty tuple is accused.

Proof. The proof is very similar to the proof of Theorem 1. Instead of n innocent and c guilty users, we now have $\binom{n}{c} < n^c$ all-innocent tuples and just one all-guilty tuple, which changes some of the numbers in γ, η, and ℓ above. We then again apply the Markov inequality with $\alpha = 1 - \sqrt{\gamma}$ and $\beta = \sqrt{\gamma}$ to obtain the given expressions for η and ℓ. □

For deterministic strategies $\vec{\theta} \in \{0, 1\}^{c+1}$, choosing the scheme parameters is much simpler. Similar to [26, Lemma 1], where it was shown that for deterministic attacks the capacity is exactly $\frac{1}{c}$, in this case, we get a code length of roughly $c \log_2 n$.

Theorem 5. *Let $\vec{\theta}$ be a deterministic attack, and let p be chosen such that $f_{Y|P}(1|p) = \frac{1}{2}$. Let ℓ and η be chosen as:*

$$\ell = \log_2 \left(\frac{n^c}{\varepsilon_1} \right), \quad \eta = \ln \left(\frac{n^c}{\varepsilon_1} \right),$$

Then, with probability $1 - \varepsilon_1$, all all-innocent tuples will not be accused, and the single all-guilty tuple will always be accused.

Proof. For deterministic attacks, we have

$$\begin{aligned} f_{Z,Y|P}(z, y|p, H_0) &= f_{Z|P}(z|p) f_{Y|Z}(y|z), \\ f_{Z,Y|P}(z, y|p, H_1) &= f_{Z|P}(z|p) f_{Y|P}(y|p). \end{aligned}$$

As a result, the score function g satisfies

$$g(z, y, p) = \begin{cases} -\ln f_{Y|P}(y|p) & \text{if } \theta_z = y; \\ -\infty & \text{if } \theta_z \neq y. \end{cases}$$

With the capacity-achieving choice of p of [26, Lemma 1], we have $f_{Y|P}(y|p) = \frac{1}{2}$ for $y = 0, 1$ leading to a score of $+\ln 2$ for a match and $-\infty$ for cases where y_i does not match the output that follows from $\vec{\theta}$ and the assumption that T is the all-guilty tuple. For $T = \mathcal{C}$, clearly, we will always have a match, so this tuple's score will always be $\ell \ln 2$, showing that this tuple is always accused.

For innocent users, $f(z, y|p, H_1) = \frac{1}{2} f(z, y|p, H_0)$ implies that in each position i, with probability $\frac{1}{2}$ this tuple's score will not be $-\infty$. So with probability $2^{-\ell}$, the tuple's score after ℓ segments will not be $-\infty$, in which case it equals

$\ell \ln 2$. To make sure that this probability is at most ε_1/n^c so that the total error probability is at most ε_1, we set $2^{-\ell} = \varepsilon_1/n^c$, leading to the result. □

Note that for deterministic attacks, any choice of $\eta_0 \leq \eta$ works just as well as choosing η; after ℓ segments all tuples will either have a score of $-\infty$ or η.

4.3 Practical evaluation

Theorem 4 does not prove that we can actually find the set of colluders with high probability, since mixed tuples (consisting of some innocent and some guilty users) also exist and these may or may not have a score exceeding η. Theorem 4 only proves that with high probability, we can find a set $\mathcal{C}'$ of c users which contains at least one colluder. Basic experiments show that in many cases, the only tuple with a score exceeding η (and thus the tuple with the highest score) is the all-guilty tuple, and so all mixed tuples have a score below η. Proving that mixed tuples indeed get a score below η is left as an open problem.

4.4 Asymptotic code lengths

To further motivate why using this joint decoder may be the right choice, the following proposition shows that at least the resulting code lengths are optimal. The proof is analogous to the proof of Theorem 2.

Theorem 6. *If $\gamma = o(1)$, then the code length ℓ of Theorem 4 scales as*

$$\ell = \frac{\log_2 n}{\frac{1}{c} I(Z; Y|P = p)} \left[1 + O(\sqrt{\gamma})\right],$$

thus asymptotically achieving an optimal scaling of the code length for arbitrary values of p.

Since the asymptotic code length is optimal regardless of p, these asymptotics are also optimal when p is optimized to maximize the mutual information in the fully informed setting.

Finally, although it is hard to estimate the scores of mixed tuples with this decoder, just like in [37], we expect that the joint decoder score for a tuple is roughly equal to the sum of the c individual simple decoder scores. So a tuple of c users consisting of k colluders and $c - k$ innocent users is expected to have a score roughly a factor k/c smaller than the expected score for the all-guilty tuple. So after computing the scores for all tuples of size c, we can get rough estimates of how many guilty users are contained in each tuple, and for instance, try to find the set $\mathcal{C}'$ of c users that best matches these estimates. There are several options for post-processing that may improve the accuracy of using this joint decoder, which are left for future work.

4.5 Fingerprinting attacks

Using Theorem 4, we can obtain exact expressions for ℓ in terms of $\vec{\theta}, p, c, n, \varepsilon_1, \varepsilon_2$. For the optimal values of p of [26, Section III.A], we can use Theorem 6 to obtain the following expressions. Note again that $\ell\left(\vec{\theta}_{\min}\right) \sim \ell\left(\vec{\theta}_{\text{all1}}\right)$.

$$\begin{aligned}
\ell\left(\vec{\theta}_{\text{int}}\right) &= 2c^2 \ln n \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c^2}\right)\right], \\
\ell\left(\vec{\theta}_{\text{all1}}\right) &= c \log_2 n \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right], \\
\ell\left(\vec{\theta}_{\text{maj}}\right) &= c \log_2 n \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right], \\
\ell\left(\vec{\theta}_{\text{coin}}\right) &= c \log_{5/4} n \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right].
\end{aligned}$$

Let us again highlight one resulting decoder in particular, the one for the interleaving attack. In general, we can rewrite the ratio inside the logarithm as follows:

$$\frac{f_{Z,Y|P}(z, y|p, H_0)}{f_{Z,Y|P}(z, y|p, H_1)} = \frac{f_{Y|Z}(y|z)}{f_{Y|P}(y|p)}.$$

For the interleaving attack, we further have $f_{Y|Z}(1|z) = \frac{z}{c}$ and $f_{Y|P}(1|p) = p$. This leads to the following joint decoder g.

$$g(z, y, p) = \begin{cases} \ln(1 - \frac{z}{c}) - \ln(1 - p) & \text{if } y = 0; \\ \ln(\frac{z}{c}) - \ln(p) & \text{if } y = 1. \end{cases}$$

This means that the joint scores are purely based on the similarities between the tuple tally z and the expected tuple tally cp for each position i. If a tuple's tally z is larger than the expected tally cp, then the score is positive if $y = 1$ and negative otherwise, while if z is smaller than cp, then the score is positive if $y = 0$ and negative otherwise. For innocent tuples, this leads to an expected score of roughly 0, while for the guilty tuple, this leads to a high (positive) expected score.

4.6 Group testing models

Similar to the above, for group testing models, we can also use Theorem 4 to obtain exact expressions for ℓ in terms of $\vec{\theta}, p, c, n, \varepsilon_1, \varepsilon_2$ with provable error bounds. For the optimal values of p of [26, Section III.B], we can use Theorem 6 to obtain the following refined expressions.

$$\begin{aligned}
\ell\left(\vec{\theta}_{\text{all1}}\right) &= c \log_2 n \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right], \\
\ell\left(\vec{\theta}_{\text{add}}\right) &= \frac{c \log_2 n}{1 - \frac{1}{2} h(r) + O(r^2)} \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right], \\
\ell\left(\vec{\theta}_{\text{dil}}\right) &= \frac{c \log_2 n}{1 - \frac{\ln 2}{2} h(r) + O(r^2)} \left[1 + O\left(\sqrt{\gamma} + \frac{1}{c}\right)\right].
\end{aligned}$$

Note that as discussed in Theorem 5, the score function for the classical model is equivalent to simply checking whether some subset of c items matches the test results, i.e., whether these would indeed have been the test results, had this subset been the set of defectives. With high probability, only the correct set of defectives passes this test.

5 Joint universal decoding

Let us now again consider the more common setting in fingerprinting where the attack strategy $\vec{\theta}$ is assumed unknown to the distributor. With the results for simple decoding in mind, and knowing that the interleaving attack is also the asymptotically optimal pirate attack in the joint fingerprinting game, we again turn our attention to the decoder designed against the interleaving attack.

5.1 The joint interleaving decoder, revisited

Similar to the setting of simple decoding, explicitly proving that this decoder achieves the uninformed capacity is not so easy. However, through a series of reductions, we can prove that this decoder is asymptotically capacity-achieving, for certain parameters ℓ and η.

Theorem 7. *The joint log-likelihood decoder designed against the interleaving attack, using the score function g defined by*

$$g(z,y,p) = \begin{cases} \ln(1-\frac{z}{c}) - \ln(1-p) & \text{if } y=0; \\ \ln(\frac{z}{c}) - \ln(p) & \text{if } y=1. \end{cases} \tag{14}$$

and the arcsine distribution encoder f_P^* of (9) together asymptotically achieve the joint capacity for the uninformed fingerprinting game.

Proof. First, the simple uninformed capacity is asymptotically equivalent to the joint uninformed capacity, which follows from results of Huang and Moulin [5] and Oosterwijk et al. [13] (and Section 2). This means that the simple universal decoder of Theorem 3 already asymptotically achieves the joint capacity. We will prove that asymptotically, the proposed universal joint decoder is equivalent to the universal simple decoder of Section 3, thus also achieving the joint uninformed capacity.

Suppose we have a tuple T of size c and suppose in some segment i there are z users who received a 1 and $c-z$ users who received a 0. For now, also assume that $p \in [\delta, 1-\delta]$ for some $\delta > 0$ that does not depend on c. In case $y=0$, the combined simple decoder score of this tuple T (using the simple universal decoder g of Section 3) would be:

$$\sum_{j \in T} S_{j,i} = z \cdot g(1,0,p) + (c-z)g(0,0,p)$$
$$= z \ln\left(1 - \frac{1}{c}\right) + (c-z)\ln\left(1 + \frac{p}{c(1-p)}\right)$$
$$\sim -\frac{z}{c} + \frac{(c-z)p}{c(1-p)} = \frac{p - z/c}{1-p}.$$

On the other hand, if we look at this tuple's joint score with the joint universal decoder g of (14), we have

$$S_{T,i} = g(z,0,p) = \ln\left(1 + \frac{p-z/c}{1-p}\right) \sim \frac{p-z/c}{1-p}.$$

Note that the last step follows from the fact that for large c, with overwhelming probability we have $z = cp + O(\sqrt{cp})$ (since Z is binomially distributed with mean cp and variance $cp(1-p)$), in which case $(p - z/c)/(1-p) = o(1)$. Combining the above, we have that $S_{T,i} \sim \sum_{j \in T} S_{j,i}$. So the joint universal decoder score for a tuple T is asymptotically equivalent to the sum of the simple universal decoder scores for the members in this tuple, if $p \in [\delta, 1-\delta]$.

Since as argued before the distribution tails $[0, \delta]$ and $[1-\delta, 1]$ are negligible for the performance of the scheme for sufficiently small δ, and since the same result holds for $y=1$, the simple and joint decoders are asymptotically equivalent. □

Note that for the uninformed setting, the simple and joint capacities are asymptotically equivalent, which allowed us to prove the result. For finite c, the joint capacity may be slightly higher than the simple capacity, but the fact that they are asymptotically the same does show that there is not as much to gain with joint decoders as there is with, e.g., joint group testing decoders, where the joint capacity is asymptotically a factor $\log_2(e) \approx 1.44$ higher than the simple capacity. And since assigning joint scores to all tuples of size c for each position i is computationally very involved (and since the resulting joint scores are very similar to the sum of simple universal scores anyway), a more practical choice seems to be to use the simple universal decoder of Section 3 instead.

6 Discussion

Let us now briefly discuss the main results in this paper and their consequences.

6.1 Simple informed decoding

For the setting of simple decoders with known collusion channels $\vec{\theta}$, we have shown how to choose the score functions g in the score-based framework, as well as how to choose the threshold η and code length ℓ to guarantee that certain bounds on the error probabilities are met. With log-likelihood decoders, we showed that these decoders achieve capacity *regardless of p and regardless of* $\vec{\theta}$. This means that no matter which pirate strategy $\vec{\theta}$ and bias p you plug in, this construction will always achieve the capacity corresponding to those choices for $\vec{\theta}$ and p. A trivial consequence is that for the optimal values of p derived in [26], one also achieves the optimal code length for arbitrary p.

6.2 Simple universal decoding

Since in fingerprinting it is usually more common to assume that $\vec{\theta}$ is unknown, we then turned our attention to the uninformed setting. We showed that the decoder designed against the interleaving attack, with the score function g given by

$$g(x,y,p) = \begin{cases} \ln\left(1 + \frac{p}{c(1-p)}\right) & \text{if } x = y = 0; \\ \ln\left(1 - \frac{1}{c}\right) & \text{if } x \neq y; \\ \ln\left(1 + \frac{1-p}{cp}\right) & \text{if } x = y = 1, \end{cases}$$

is actually a universal decoder and achieves the uninformed fingerprinting capacity. We also showed how the proposed universal decoder is very similar to both Oosterwijk et al.'s decoder h and an approximation of Moulin's empirical mutual information decoder m for $0 \ll p \ll 1$ by

$$c \cdot g(x,y,p) \sim h(x,y,p) \sim n \cdot m(x,y,p),$$

and we highlighted the differences between these decoders for $p \approx 0, 1$. We argued that the proposed decoder g is the most natural choice for a universal decoder (motivated from the Neyman-Pearson lemma) and that it has some practical advantages compared to Oosterwijk et al.'s decoder h, such as finally being able to get rid of the cut-offs δ on the density function f_P.

6.3 Joint informed decoding

In Sections 4 and 5, we then turned our attention to joint decoders, which have the potential to significantly decrease the required code length ℓ at the cost of a higher decoding complexity. We considered a natural generalization of the simple decoders to joint decoders and argued that our choice for the joint score functions g seems to be optimal. There are still some gaps to fill here, since we were not able to prove how scores of mixed tuples (tuples containing some innocent and some guilty users) behave and whether their scores also stay below η with high probability. On the other hand, for deterministic attacks, it is quite easy to analyze the behavior of these decoders, and for arbitrary attacks, we did show that the code lengths have the optimal asymptotic scaling.

6.4 Joint universal decoding

Finally, for the uninformed setting with joint decoders, we proved that the joint interleaving decoder achieves the joint uninformed capacity. Since the joint uninformed capacity is asymptotically the same as the simple uninformed capacity, and since joint decoding generally has a much higher computational complexity than simple decoding, this decoder may not be as practical as the proposed simple universal decoder.

7 Conclusions

Finally, let us finish by mentioning some open problems which are left for future work.

7.1 Analyzing the simple universal decoder

While in Section 3 we showed that the new simple universal decoder achieves capacity and how one should roughly choose the code length ℓ and threshold η, we did not provide any provable bounds on the error probabilities for the uninformed setting. For earlier versions of Tardos' scheme various papers analyzed such provable bounds [6, 7, 9, 13, 29] and a similar analysis could be done for the log-likelihood decoder designed against the interleaving attack. Perhaps such an analysis may once and for all establish the best way to choose the scheme parameters in universal fingerprinting for large parameters c and n.

7.2 Dynamic fingerprinting and adaptive group testing

Although this paper focused on applications to the "static" fingerprinting game, in some settings, the feedback Y may be obtained in real time. For instance, in pay-tv, pirates may try to duplicate a fingerprinted broadcast in real time, while in group testing, it may sometimes be possible to do several rounds of group testing sequentially. The construction of [40] can trivially be applied to the decoders in this paper as well to build efficient dynamic fingerprinting schemes with the same asymptotics for the code lengths but where (i) the order terms in ℓ are significantly smaller; (ii) one can provably catch *all* pirates regardless of the (asymmetric) pirate strategy; and (iii) one does not necessarily need to know (a good estimate of) c in advance [40, Section V]. An important open problem remains to determine the dynamic uninformed fingerprinting capacity, which may prove or disprove that the construction of [40] combined with the universal decoder g of Section 3 is asymptotically optimal.

7.3 Non-binary codes in fingerprinting

In this work, we focused on the binary case of $q = 2$ different symbols for generating the code $\mathcal{X}$, but in (universal) fingerprinting, it may be advantageous to work with larger alphabet sizes $q > 2$, since the code length decreases linearly with q [38, 39]. This generalization to q-ary alphabets was considered in, e.g., [9, 13, 37–39]. For the results in this paper, we did not really use that we were working with a binary alphabet, so it seems a straightforward exercise to prove that the q-ary versions of the log-likelihood decoders also achieve the q-ary capacities. A harder problem seems to be to actually compute these capacities in the various informed settings, since the maximization problem involved in computing these capacities then transforms from a one-dimensional optimization problem to a $(q - 1)$-dimensional optimization problem.

Endnotes

[1] In fingerprinting, a common generalization is to assume that the entries of the code words come from an alphabet of size $q \geq 2$, but in this paper, we restrict our attention to the binary case $q = 2$.

[2] In those cases, attacks exist guaranteeing you will not catch more than one colluder, such as the "scapegoat" strategy [40].

[3] To be precise, for convenience, we have scaled g by a factor $(c \ln 2)$, and so also η should be scaled by a factor $(c \ln 2)$.

[4] When p_i is not fixed and is drawn from a continuous distribution function f_P, the empirical probabilities considered below do not make much sense, as each value of p_i only occurs once. In that case, one could, e.g., build a histogram for the values of p and compute empirical probabilities for each bin or discretize the distribution function for p [8, 10].

Competing interests

The author declares that he has no competing interests.

Acknowledgements

The author is grateful to Pierre Moulin for his comments during the author's visit to Urbana-Champaign that lead to the study of the empirical mutual information decoder of Theorem 2 and that eventually inspired work on this paper. The author further thanks Jeroen Doumen, Teddy Furon, Jan-Jaap Oosterwijk, Boris Škorić, and Benne de Weger for the valuable discussions and comments regarding earlier versions of this manuscript.

Received: 21 July 2015 Accepted: 16 December 2015
Published online: 04 January 2016

References

1. R Dorfman, The detection of defective members of large populations. Ann. Math. Stat. **14**(4), 436–440 (1943)
2. D Boneh, J Shaw, Collusion-secure fingerprinting for digital data. IEEE Trans. Inf. Theory. **44**(5), 1897–1905 (1998). doi:10.1109/18.705568
3. G Tardos, in *STOC*. Optimal probabilistic fingerprint codes, (2003), pp. 116–125. doi:10.1145/780542.780561
4. E Amiri, G Tardos, in *SODA*. High rate fingerprinting codes and the fingerprinting capacity (SIAM, Philadelphia, PA, USA, 2009), pp. 336–345
5. Y-W Huang, P Moulin, On the saddle-point solution and the large-coalition asymptotics of fingerprinting games. IEEE Trans. Inf. Forensics Secur. **7**(1), 160–175 (2012). doi:10.1109/TIFS.2011.2168212
6. O Blayer, T Tassa, Improved versions of Tardos' fingerprinting scheme. Designs Codes Crypt. **48**(1), 79–103 (2008). doi:10.1007/s10623-008-9200-z
7. T Laarhoven, B de Weger, Optimal symmetric Tardos traitor tracing schemes. Designs Codes Crypt. **71**(1), 83–103 (2014). doi:10.1007/s10623-012-9718-y
8. K Nuida, S Fujitsu, M Hagiwara, T Kitagawa, H Watanabe, K Ogawa, H Imai, An improvement of discrete Tardos fingerprinting codes. Designs Codes Crypt. **52**(3), 339–362 (2009). doi:10.1007/s10623-009-9285-z
9. B Skoric, S Katzenbeisser, MU Celik, Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Designs Codes Crypt. **46**(2), 137–166 (2008). doi:10.1007/s10623-007-9142-x
10. T Laarhoven, B de Weger, in *IH&MMSec*. Discrete distributions in the Tardos scheme, revisited, (2013), pp. 13–18. doi:10.1145/2482513.2482533
11. E Abbe, L Zheng, Linear universal decoding for compound channels. IEEE Trans. Inf. Theory. **56**(12), 5999–6013 (2010). doi:10.1109/TIT.2010.2080910
12. P Meerwald, T Furon, Toward practical joint decoding of binary Tardos fingerprinting codes. IEEE Trans. Inf. Forensics and Secur. **7**(4), 1168–1180 (2012). doi:10.1109/TIFS.2012.2195655
13. B Oosterwijk, J-J Skoric, J Doumen, A capacity-achieving simple decoder for bias-based traitor tracing schemes. IEEE Trans. Inf. Theory. **61**(7), 3882–3900 (2015). doi:10.1109/TIT.2015.2428250
14. A Charpentier, F Xie, C Fontaine, T Furon, in *SPIE Media Forensics and Security*. Expectation maximization decoding of Tardos probabilistic fingerprinting code, (2009), pp. 1–15. doi:10.1117/12.806034
15. T Furon, L Pérez-Freire, in *MMSec*. EM decoding of Tardos traitor tracing codes, (2009), pp. 99–106. doi:10.1145/1597817.1597835
16. T Laarhoven, in *WIFS*. Dynamic traitor tracing schemes, revisited, (2013), pp. 191–196. doi:10.1109/WIFS.2013.6707817
17. P Meerwald, T Furon, in *ICASSP*. Group testing meets traitor tracing, (2011), pp. 4204–4207. doi:10.1109/ICASSP.2011.5947280
18. A Sebő, On two random search problems. J. Stat. Plan. Infer. **11**(1), 23–31 (1985). doi:10.1016/0378-3758(85)90022-9
19. AG D'yachkov, VV Rykov, Bounds on the length of disjunctive codes. Probl. Peredachi Informacii. **18**(3), 7–13 (1982)
20. AG D'yachkov, VV Rykov, AM Rashad, Superimposed distance codes. Probl. Control Inf. Theory. **18**(4), 237–250 (1989)
21. V Atia, GK Saligrama, Boolean compressed sensing and noisy group testing. IEEE Trans. Inf. Theory. **58**(3), 1880–1901 (2012). doi:10.1109/TIT.2011.2178156
22. M Cheraghchi, A Hormati, A Karbasi, M Vetterli, Group testing with probabilistic tests: Theory, design and application. IEEE Trans. Inf. Theory. **57**(10), 7057–7067 (2011). doi:10.1109/TIT.2011.2148691
23. T Laarhoven, in *ALLERTON*. Efficient probabilistic group testing based on traitor tracing, (2013), pp. 1358–1365. doi:10.1109/Allerton.2013.6736699
24. D Sejdinovic, O Johnson, in *ALLERTON*. Note on noisy group testing: asymptotic bounds and belief propagation reconstruction, (2010), pp. 998–1003. doi:10.1109/ALLERTON.2010.5707018
25. P Moulin, Universal fingerprinting: capacity and random-coding exponents, 1–69 (2008). arXiv:0801.3837 [cs.IT]
26. T Laarhoven, Asymptotics of fingerprinting and group testing: tight bounds from channel capacities. IEEE Trans. Inf. Forensics and Secur. **10**(9), 1967–1980 (2015). doi:10.1109/TIFS.2015.2440190
27. J Neyman, ES Pearson, On the problem of the most efficient tests of statistical hypotheses. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **231**(694-706), 289–337 (1933). doi:10.1098/rsta.1933.0009
28. L Pérez-Freire, T Furon, in *WIFS*. Blind decoder for binary probabilistic traitor tracing codes, (2009), pp. 46–50. doi:10.1109/WIFS.2009.5386486
29. S Ibrahimi, B Skoric, J-J Oosterwijk, Riding the saddle point: asymptotics of the capacity-achieving simple decoder for bias-based traitor tracing. EURASIP J. Inf. Secur. **1**(12), 1–11 (2014). doi:10.1186/s13635-014-0012-6
30. W Berchtold, M Schäfer, in *MMSec*. Performance and code length optimization of joint decoding Tardos fingerprinting, (2012), pp. 27–32. doi:10.1145/2361407.2361412
31. M Desoubeaux, C Herzet, W Puech, GL Guelvouit, in *MMSP*. Enhanced blind decoding of Tardos codes with new MAP-based functions, (2013), pp. 283–288. doi:10.1109/MMSP.2013.6659302
32. T Furon, M Desoubeaux, in *WIFS*. Tardos codes for real, (2014), pp. 24–29. doi:10.1109/WIFS.2014.7084298
33. M Kuribayashi, in *IH*. Bias equalizer for binary probabilistic fingerprinting codes, (2013), pp. 269–283. doi:10.1007/978-3-642-36373-3
34. DR Stinson, T van Trung, R Wei, Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. J. Stat. Plan. Infer. **86**(2), 595–617 (2000). doi:10.1016/S0378-3758(99)00131-7
35. CL Chan, PH Che, S Jaggi, V Saligrama, in *ALLERTON*. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms, (2011), pp. 1832–1839. doi:10.1109/Allerton.2011.6120391
36. CL Chan, S Jaggi, V Saligrama, S Agnihotri, in *ISIT*. Non-adaptive group testing: Explicit bounds and novel algorithms, (2012), pp. 1837–1841. doi:10.1109/ISIT.2012.6283597
37. J Oosterwijk, J-J Doumen, T Laarhoven, in *SPIE Media Watermarking, Security, and Forensics*. Tuple decoders for traitor tracing schemes, (2014), pp. 1–21. doi:10.1117/12.2037659
38. D Boesten, B Skoric, in *IH*. Asymptotic fingerprinting capacity for non-binary alphabets, (2011), pp. 1–13. doi:10.1007/978-3-642-24178-9_1
39. Y-W Huang, P Moulin, in *ISIT*. On fingerprinting capacity games for arbitrary alphabets and their asymptotics, (2012), pp. 2571–2575. doi:10.1109/ISIT.2012.6283982
40. T Laarhoven, J Doumen, P Roelse, B Skoric, B de Weger, Dynamic Tardos traitor tracing schemes. IEEE Trans. Inf. Theory. **59**(7), 4230–4242 (2013). doi:10.1109/TIT.2013.2251756