DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

A Novel Attack Model for Collusion Secure Fingerprinting Codes

Marcel Schäfer[†,1]***, Waldemar Berchtold***[†,2]***, Teetje Stark***[*,3]***, Nils Reimers***[•,4]***, Martin Steinebach***[†,5]

[†] ***Fraunhofer-Institute SIT, Darmstadt, Germany***
[*] ***Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany***
[•] ***Ubiquitous Knowledge Processing Lab – Department of Computer Science, Technische Universität Darmstadt, Darmstadt, Germany***

[1]***marcel.schaefer@sit.fraunhofer.de,*** [2]***waldemar.berchtold@sit.fraunhofer.de,*** [3]***teetje.stark@fu-berlin.de,***
[4]***reimers@ukp.informatik.tu-darmstadt.de,*** [5]***martin.steinebach@sit.fraunhofer.de***

Abstract

Probabilistic or bias-based fingerprinting codes to counter collusion attacks are applied to enhance the security of transaction watermarking applications. Into every media copy to be distributed is embedded a unique fingerprint via watermarking techniques. With it a distributor is able to trace back unauthorized redistributed versions to its source with a high probability even if the media was subject to a collusion attack. The seminal Tardos fingerprinting codes and all its derivatives rely on position independent fingerprints. Hence in most work it is assumed that the attackers also rely on the position independence when creating the media forgery containing a manipulated fingerprint. However, they need not follow this assumption. In this work we present a novel iterative attack model that does not rely on the position independence but could be applied in practice. The corresponding attacks iteratively adapt the manipulated fingerprint with the intention to maximally reduce their accusation scores in order to escape the accusation. For practical collusion sizes, the attacks show better performance than other attacks typically discussed in literature. In other words, the attacks result in manipulated fingerprints that lead to higher error rates of the fingerprinting scheme, compared to the attacks discussed in literature.

1 Introduction

In the last two decades digital watermarking has become a grown up alternative to digital rights management (DRM) methods. Providing the possibility of embedding a unique identifier – the watermark message – into every media copy to be sold, potential dishonest customers of watermarked media content are discouraged from unauthorized redistribution, because the unique identifier allows tracing back to them. Research has driven the performance of watermarking algorithms so far that most common media operations and also attacks with the intention to destroy the watermark cannot prevent detecting the correct watermark unless the quality of the media cover is significantly degraded. Collusion attacks on the other hand still pose a great risk. To conduct a collusion attack several customers who purchased the same media content – each containing a unique watermark message – *collude* and compare their media copies. Thereby they detect differences and assume these to be the differing watermark information. Thus, they are able to manipulate the media copy only at these positions and within their difference range and hence create a media copy of the same quality, yet containing a watermark that is a mixture of their watermark information, preventing a clear identification. In recent years, to counter collusion attacks, various collusion secure fingerprinting codes have been introduced These are mathematical codes that are embedded into media copies as watermarking messages. After an unauthorized media file is retrieved and the watermark detection process extracts the fingerprint contained, a tracing algorithm is run utilizing the probabilistic information from the fingerprint generation to calculate scores regarding the suspiciousness of the fingerprints in the database. Given that, the most suspicious fingerprint(s) can be accused.

In literature most effort is spent on the asymptotic (very large collusions) optimal code. To achieve this, the goal is to make the best of the two-party max-min game between code designers and attackers. The encoders try to get the most information about the colluders, whereas these try to disclose the least possible information about themselves. The so called fingerprint capacity, [27], is where both parties have their equilibrium.

The seminal work of probabilistic or bias-based fingerprinting codes is the well known *Tardos Codes* introduced by Tardos [35] proving that a fingerprinting scheme must satisfy $m \propto c^2 \ln(n\varepsilon_1^{-1})$ for a large number of fingerprints n, with code length m, number of colluders c and upper bound on the probability of accusing a specific innocent ε_1. *Modern* fingerprinting codes typically rely on the Tardos codes, proposing optimizations of the fingerprint generation or the tracing algorithm, e.g. [37], [3], [21], [34], [28], [30], [16], [29], [9], [1], [13]) [36], [17], [10], [31], [32], [23], [8], [24].

Tardos suggested the arcsine distribution for generating binary fingerprinting codes already in [35], later this choice was proven to achieve the capacity bounds independently in [1] and [13]. With the interleaving attack, for which colluders substitute to the colluded fingerprint uniformly at random, as the asymptotic optimal attack – proven in [14] – the last years yield several special decoders against this attack [8], [31], [20], as to be the best *defense*. The first aiming at practical parameters, the latter two achieving capacity, that is achieving asymptotic optimal code lengths for binary fingerprints of $m \sim 2c^2 \ln(n\varepsilon_1^{-1})$. These decoders are so-called single decoders as the scores are calculated

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

independently for each fingerprint.

Decoders using a joint decoding approach, which means the scores are no longer calculated per fingerprint but per tuple of fingerprints [24], [22], [2], are agreed upon to perform better than single decoders, but the increment in performance goes along with an increase in computational complexity. Joint decoding is no longer considered here.

Most fingerprinting codes follow Boneh and Shaw's so called marking assumption [6], saying that the colluders are only able to modify detectable positions, i.e. positions where their watermarked media copies differ from each other due to the individual watermark information embedded. Besides, a typical assumption model found in literature within the marking assumption is the location independence of the attack strategy, i.e. the memoryless collusion channel, e.g. [38].

The reason for the assumption that the colluders restrict themselves to a memoryless attack strategy is that the code generations as well as the score functions work completely position-symmetric, so it might be disadvantageous for the attackers to deviate from this symmetry. Moreover, it significantly simplifies the proofs for the security of the corresponding fingerprinting scheme. Also, Moulin shows in [25] that enforcing this restriction does not change the fingerprint capacity asymptotically, which means in case the number of attackers is tending to infinity. However, in the real world the colluders are not restricted to this assumption. Instead they could apply attack strategies for which each symbol is chosen also with respect to the symbols observed on all other detectable positions. To the best of our knowledge, the behavior of the state of the art fingerprinting codes for this new attack model has not been considered yet.

In this work we re-define stateful attacks, i.e. attacks that do not need to follow a memoryless collusion channel. With it we propose a new kind of collusion attack and also explain explicit examples. Results show a similar but nevertheless improved (with regards to the attackers) behavior of the new attacks compared to the commonly tested stateless or memoryless attacks such as interleaving or minority vote attack.

The paper is structured as follows: In section 2 we describe the three stages specific for fingerprinting schemes, the fingerprint generation, the attack/collusion channel covering two different attack strategies and the tracing algorithm with respect to four different decoders recently discussed in literature. section 3 specifies the basic idea that lead to this work, the stateful attack strategies. This includes a general description of this yet unconsidered type of attack as well as two example strategies detailed afterwards. The major part of this work takes up section 4 containing a digest of the evaluation necessary to classify the new attacks compared to the known ones and with respect to the different decoders and tracing goals. These lead to a partition of the section into the evaluation of the 'detect one' scenario, i.e. tracing one colluder, and into the evaluation of the 'detect many' scenario, i.e. tracing as many suspicious users as possible. section 5 sums up the important findings and concludes this paper.

2 Fingerprinting Scheme and Principles

The fingerprinting scheme consists of the fingerprint generation process, the fingerprint tracing algorithm and in between a complete watermarking scheme (embedding, attacking, detecting/extracting). In this work we will only focus on the processes specific for fingerprinting. This section provides the corresponding background knowledge required for the evaluation section 4.

2.1 Preliminaries

Before the generation of the fingerprints some fingerprinting parameters need to be set. The applier needs to select the desired or acceptable upper bounds of the errors that can occur. In a fingerprinting scheme, two kinds of errors are discussed. The false positive error describes the event in which an innocent-fingerprint is wrongly suspected of having partaken in the collusion. The corresponding upper bound is commonly denoted as ε_1. On the other hand, the event in which none of the colluder-fingerprints can be identified and suspected is referred to as false negative error. Its upper bound is denoted as ε_2. Since the error rates depend on each other in some way, every fingerprinting code has to provide realistic values for both. Since it is worse to accuse an innocent-fingerprint than to accuse no one, the false positive error rate obviously needs to be very small, whereas a false negative error rate of 0.5 sometimes is sufficient. In addition, the expected threat level, i.e. the maximum number of colluders c_0 the code needs to be resistant to, has to be determined. The actual number of colluders c must not exceed c_0 to provably stay within the selected error bounds.

2.2 Fingerprint generation

The generation of the fingerprints is according to the Tardos Codes, i.e. using the arcsine distribution [35] and with optimized parameter selection according to [21]. Therewith an $n \times m$ matrix X is generated, with n denoting the number of fingerprints and m refers to the minimum code length the scheme promises provable correctness for. Ergo the j^{th} row of the matrix corresponds to the fingerprint which is later embedded in the copy that is released to customer $j \in \{1,...,n\}$. The entries X_{ji} of matrix X are generated in two steps: First, the distributor picks m independent random numbers $\{p_i\}_{i=1}^{m}$ according to the arcsine distribution over the interval $p_i \in [t, 1-t]$, with a certain cutoff t. Second, the matrix X is filled, by picking each entry X_{ji} independently from the binary alphabet $\{0,1\}$ according to $\mathbb{P}[X_{ji} = 1] = p_i$. Note that while there are generalizations to larger q-ary alphabets, for instance [36] and [4], we restrict ourselves to the binary case $q = 2$ as implied above. This is because most watermarking algorithms embed binary messages. Hence fingerprints with higher alphabets would have to be downscaled to binary messages anyway thus questioning the value of a larger alphabet.

2.3 Collusion attacks

A group of malicious customers, also called colluders or pirates, can execute different collusion attacks in order to avoid detection by the tracing algorithm. In this work we follow the marking assumption saying that the colluders are only able to modify detectable positions, i.e. positions where their watermarked media copies differ from each other due to the individual watermark information embedded. Though most approaches follow this assumption, there exist several extensions and accentuations of this assumption, e.g. [5], [28], but as all relaxations of this assumption complicate the code construction and tracing, and its affect on the whole scheme is very depending on the basic watermarking algorithm, we will restrict this work to the core marking assumption as given above.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

Apart from that, a typical assumption model found in literature within the marking assumption is the location independence of the attack strategy.

Definition 1 (Stateless attack model/memoryless channel) *In an attack model that is location independent, the not-necessarily deterministic output y_i of the position i only depends on the symbols $X_{j_1 i},...,X_{j_c i}$ of the colluder fingerprints $X_{j_1},...,X_{j_c}$ of that particular position i. It is independent of other positions or any other output y_k, with $k \neq i$. We will refer to this attack model as stateless attack model.*

The reason for the assumption that the colluder restrict themselves to a stateless attack strategy is that the code generation as well as the score function work completely position-symmetric, so it might be be disadvantageous for the attackers to deviate from this symmetry. Moreover, it significantly simplifies the proofs for the security of the corresponding fingerprinting scheme. Also, Moulin shows in [25] that enforcing this restriction does not change the fingerprint capacity asymptotically.

Let $C = \{j_1,...,j_c\} \subseteq \{1,...,n\}$ denote a collusion of size c with corresponding fingerprints $X_{j_1},...,X_{j_c}$. Be X_C the corresponding matrix of colluder-fingerprints. The colluders create a forged fingerprint $y = \rho(X_C)$ according to a (possibly non-deterministic) strategy ρ with the constraint that $y_i = X_{j_1 i}$ if $X_{j_1 i} = ... = X_{j_c i}$ (marking assumption). For stateless attack strategies the strategy ρ does not depend on the column index i and the same strategy is applied to all columns. Therefore the different symbols of the forged fingerprint y are generated by $y_i = \rho(X_{C,i})$ where $X_{C,i}$ denotes the i-th column of X_C.

Usually it is assumed that the colluders want to participate equally in the forgery. If a colluder is idle, there is no hope in identifying this colluder. We therefore assume that the strategy is invariant under permutation of the colluder identities, what is common sense in literature.

Let λ_i be the number of ones in the column $X_{C,i}$, with $0 \leq \lambda_i \leq c$. The attack strategy ρ can be parameterized by a set of probabilities $\theta = (\theta_{\lambda_i})_{0 \leq \lambda_i \leq c}$ with $\mathbb{P}[y_i = 1] = \theta_{\lambda_i}$. The marking assumption enforces $\theta_0 = 0$ and $\theta_c = 1$.

- *Interleaving attack*: For the interleaving attack, an index $k \in C$ is selected uniformly at random and $y_i = X_{k,i}$. Using the above notation, the interleaving attack can be parameterized as
$$\theta_{\lambda_i} = \lambda_i / c.$$
- *Minority Vote attack*: For the minority vote attack, the colluders choose the least common symbol. In case of a tie, a symbol is selected uniformly at random. The attack can be parameterized as
$$\theta_{\lambda_i} = \begin{cases} 1 & \text{if } 0 < \lambda_i < \frac{1}{2}c \text{ or } \lambda_i = 1 \\ \frac{1}{2} & \text{if } \lambda_i = \frac{1}{2}c \\ 0 & \text{if } \frac{1}{2}c < \lambda_i < c \text{ or } \lambda_i = 0. \end{cases}$$

In [15] Huang and Moulin show that the interleaving attack is the strongest attack in an asymptotic sense. On the contrary, in many practical settings the minority vote attack effectuated the highest error rates. For this reason we selected these two attack strategies to be compared to the stateful strategies proposed in section 3.

2.4 Tracing algorithm

Once a forgery has been found, the distributor will try to identify the customers who partook in crafting it. Therefore he employs a tracing algorithm calculating a score for every fingerprint (single decoding) where higher scores correspond to a higher likelihood of having partaken in the collusion. Several decoders have been proposed in recent years. Some very famous or recently discussed in literature are described in the following.

- **The Škorić decoder** Among the most famous is the decoder featuring the symmetric score function of Škorić et al. [36], which improves on the score function originally introduced by Tardos [35]. It computes the score of user j at position i according to
$$S_{j,i} = \begin{cases} \sqrt{\frac{p_i}{1-p_i}} & \text{if } X_{ji} = y_i = 0 \\ -\sqrt{\frac{1-p_i}{p_i}} & \text{if } X_{ji} = 1,\ y_i = 0 \\ -\sqrt{\frac{p_i}{1-p_i}} & \text{if } X_{ji} = 0,\ y_i = 1 \\ \sqrt{\frac{1-p_i}{p_i}} & \text{if } X_{ji} = y_i = 1. \end{cases}$$
- **The Oosterwijk decoder** Subsequently it became obvious that the Škorić decoder operates below capacity, that is it could be further enhanced. In 2013 Oosterwijk et al. reported their capacity-achieving decoder in [30] that is tailored against the interleaving attack. Scores are computed as follows:
$$S_{j,i} = \begin{cases} \frac{p_i}{1-p_i} & \text{if } X_{ji} = y_i = 0 \\ -1 & \text{if } X_{ji} \neq y_i \\ \frac{1-p_i}{p_i} & \text{if } X_{ji} = y_i = 1. \end{cases}$$
- **The Laarhoven decoder** Additionally Laarhoven suggested a similar decoder [20], which in many cases features lower error rates and which does not need a cutoff parameter any more, although [8] reports that it suffers performance once no cutoff is used. While asymptotically scores equal c_0-times the *Oosterwijk* ones, for small collusion sizes we get again a Gaussian-shaped distribution of scores according to
$$S_{j,i} = \begin{cases} \ln\left(1 + \frac{p_i}{c_0(1-p_i)}\right) & \text{if } X_{ji} = y_i = 0 \\ \ln\left(1 - \frac{1}{c_0}\right) & \text{if } X_{ji} \neq y_i \\ \ln\left(1 + \frac{1-p_i}{c_0 p_i}\right) & \text{if } X_{ij} = y_i = 1. \end{cases}$$
- **The Meerwald-Furon decoder** Going in a different direction Meerwald and Furon proposed an iterative decoder [22] that is based on a worst-case estimate of the collusion channel θ and then computes the log-likelihood ratio
$$S_{j,i} = \ln\left(\frac{\mathbb{P}(y_i \mid X_{ji}, p_i, \theta)}{\mathbb{P}(y_i \mid p_i, \theta)}\right).$$
We choose to run this decoder with only a single iteration, losing the advantage of iterative decoding in a trade-off for lower computational complexity. Besides, the results of a single iteration are quite impressive themselves. Also for complexity reasons we choose to employ a custom estimator for the estimation of the collusion channel that performs slightly worse than the proposed estimator for large c or the minority vote attack.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

2.5 Accusation

We distinguish the tracing scheme depending on their intent by two different kinds. First, there is the 'detect one' scenario in which the decoder is interested solely in finding a single pirate. Second, a decoder may attempt to accuse as many pirates as possible. This setting is called 'detect many' scenario. Accusation in the 'detect one' scenario is straightforward to blame the user that has the highest score. Residing in the 'detect many' setting, things become a little more complex. The decoder wants to accuse as many colluders as possible, while maintaining a reasonable bound on the false positives. Therefore he sets a suitable threshold Z and accuses every user exceeding it. However the choice of such a threshold is a difficult challenge. Tardos set a predefined value independent of the actual attack. While this approach can usually bound the false positive rate, many times no colluder can be charged. As such unfortunate cases are sought to be avoided, different approaches have been proposed, e.g. [16], [33], [19]. In this work we restrict ourselves to the threshold calculation based on rare event analysis proposed in [12] and [7], which was already applied in e.g. [22] and [24]. Doing otherwise would blow up the evaluation section tremendously.

The underlying idea of rare event analysis builds on the fact that any fingerprint that has not been assigned to a user did not partake in the forging process. Therefore the likelihood for an innocent score to surpass a given threshold can be estimated from newly generated fingerprints. Such adaptive thresholds perform significantly better in practice (e.g. [8]) than predetermined ones.

3 Stateful Attacks

As presented in the previous section the decoder calculates scores for every fingerprint over all positions of the manipulated fingerprint y in order to weigh a user's suspiciousness. Due to the scoring function's location independence the score S_j for user j can be separated into position-wise scores $S_{j,d}$ calculated over all his detectable positions $d \in D$, i.e. the positions that the colluders could detect when comparing their copies, and position-wise scores $S_{j,u}$ calculated over his undetectable positions $u \in U$, i.e. where the colluders' copies show the same information. It holds $S_j = \sum_{d\in D} S_{j,d} + \sum_{u\in U} S_{j,u}$. As the score over the undetectable positions $\sum_{u\in U} S_{j,u} = S_U$ is equal for all colluders, the best attack strategy for a collusion, if they want none of their members caught, is a strategy that outputs a fingerprint y minimizing the probability that the largest score S_{max} of the colluder-fingerprints exceeds the accusation threshold Z:

$$\min(\mathbb{P}[S_{max} > Z]) = \min(\mathbb{P}[S_{max,D} > Z - S_U])$$

As the colluders do not know the probabilities that were used for the generation of the fingerprints, they cannot compute $S_{max,D}$ precisely. However, for instance using the maximum-likelihood method, the colluders may guess the probability for each (detectable) position from the observed symbols of those positions. The maximum-likelihood method results to the estimation $p'_i = \lambda_i / c$ and consequently

$$S'_{j,D} = \sum_{d\in D} g(X_{jd}, y_d, p'_d)$$

seems to be a decent estimate, where g denotes the score function in use. The value

$$y' = \arg\min_y(\max(S'_{j,D}))$$

therefore appears to be a reasonable candidate to minimize the expectation value of $S_{max,D}$.

With respect to these considerations, the goal is to design an attack strategy that finds valuable approximations of y' in an efficient way. However, this requires the colluders to deviate from the location independence. Recall that the reason to assume that a collusion would adhere to the location independence was not inherent but rather a consideration that it may hurt them, while definitely not benefiting them asymptotically. Consequently a collusion may well choose to resort to location dependent forging if they see an advantage.

Definition 2 (Stateful attack model) *In an attack model that is location dependent, in order to choose the symbol of the current position i for the manipulated colluded fingerprint y as the result of the collusion attack, the colluders may use the information about the symbols , $k = 1,...,m$ taken for the other positions $k \neq i$ as well. This attack model is referred to as stateful attack model.*

We propose a simple and practical stateful attack strategy referred to as 'greedy attack', as it applies a *greedy* strategy to generate the manipulated fingerprint y. Additionally we present another similar stateful attack strategy ('combinatorial attack'), which sometimes exceeds our greedy approach, yet is not applicable in practice. In the following a detailed description of both greedy and combinatorial attack is given.

3.1 Greedy attack

For this stateful attack strategy the colluders aim to minimize their accusation scores, i.e. to minimize the expectation value of $S_{max,D}$. More precisely, they approximate the column probabilities p_i, $i = 1,...,m$, of the fingerprint matrix X by analyzing the corresponding matrix of colluder-fingerprints X_C. With the approximated values for p, the colluders compute all their expected accusation scores up to position $i-1$. For the selection of the symbol in the current position i of the manipulated fingerprint y, they choose whether to put a '1' or a '0' according to the symbol that minimizes the expected accusation score highest after their choice. Afterwards they try to further improve (that is decrease) their maximum score by trying to flip the bits of y until this yields no further reduction.

To compute the expected scores the colluders also have to guess which decoder is tracing them. To avoid confusion, this guessed decoder, i.e. the score function the colluders select to create their expected scores during the attack, is from now on referred to as *collusion-decoder* whereas the actual decoder that is used to calculate the scores for all fingerprints during the tracing algorithm is denoted as *tracing-decoder*. From the decoders listed in section 2 we exclude the Meerwald-Furon decoder from valid expected collusion-decoders, since the tracer's estimate of θ depending on the colluders' estimate of the tracer's estimate of the colluders' strategy is not only very time-consuming to compute but also small-scale tests suggest it is working out rather badly

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

for the collusion as error rates are very inconsistent and oftentimes low.

Note that in a watermarking scenario in praxis, the colluders have no possibility to tell what the value of the symbol they have actually is. With respect to watermarking the colluders can only detect different media information. We admit that for this attack to be applicable in practice, the colluders need to be able to associate the detected different media information to the correct watermark/fingerprint position, which is not trivially realistic for every watermarking algorithm. Given this, however, they only need to be able to count how many of them have equal watermark information associated to one symbol ('1' or '0') compared to how many have watermark information associated to the other symbol. Therewith they can guess the probabilities p_i that are required for this attack.

3.2 Combinatorial attack

This stateful attack tries to exploit the symmetry of the scoring function. For if the fingerprints' values at two positions $k \neq l$ are equal for all colluders, then we assume $p_k = p_l$ and choosing '1' in position k and '0' in position l in the forgery y leads to expected scores of 0 for each colluder j at those two positions (assuming the scoring function is indeed perfectly symmetric like the symmetric Tardos scoring function in [36]). The same applies if the fingerprints' values at two positions differ for each colluder. Then choosing the same symbol leads to an expected score of 0 for every colluder for both positions. For those positions whose value can not be chosen this way we resort to the greedy attack. Unfortunately it relies heavily on the symmetry of the scoring function in use. However, decoders tend to be at least slightly symmetric as they try to discriminate by decreasing innocents' scores and increasing pirates' scores.

More importantly, this attack is limited by the requirement to distinguish symbols at *different* positions. Hence, this attack may not be applicable in practice but still serves as reminder that stateful attacks can not simply be discarded for non-asymptotic (that is practical) purposes.

4 Evaluation and Results

We divide our results into 'detect one' and 'detect many' scenario and present them separately. In the 'detect one' scenario an error occurs if the score of an innocent-fingerprint is larger than the maximum score of the colluder-fingerprints. Hence, an error in this 'detect one' scenario would automatically result in either a false positive or a false negative error in the 'detect many' scenario. Therefore the error probability of a fingerprinting scheme with the 'detect one' tracing algorithm is a lower bound for the error probability of the same scheme with the 'detect many' tracing algorithm. Because of this, studying the 'detect one' scenario is insofar interesting, as it gives a first impression of the best possible error rate that could be achieved by using a simple tracing-decoder. Afterwards we focus on the more common 'detect many' scenario, which provides further insight to the effects of stateful attacks on the error rates and portion of accused users.

4.1 Evaluation in a 'detect one' scenario

One advantage of the 'detect one' accusation scenario is the independence of different threshold calculations (e.g. [36], [16], [33], [22]). This permits drawing valid conclusions on the attacks' effectiveness that cannot be made void by newly proposed thresholding techniques. We first restrict our evaluation to the Škorić (collusion- and tracing-) decoder, as its score function is position and symbol symmetric and the stateful attack strategies introduced in section 3 are initially based on this symmetry. Afterwards we compare the different decoders described in section 2 with regards to their vulnerability to the stateful attacks as well as interleaving and minority vote attack.

Results for the Škorić decoder: Depicted in Figure 1, we tried to analyze the error rates for varying collusion sizes c and collusion strategies in order to gain insight as to the importance of these. The minority vote attack is depicted as we found it to be (on average) the strongest stateless attack for these parameter settings. Proven to be asymptotically optimal [14], we also show results of the interleaving attack, although it lags behind significantly. Furthermore we show the performance of our proposed stateful attacks. We restrict our results here to the Škorić collusion- and tracing-decoder according to [36], due to the combinatorial attack's reliance on the symmetry of the score function. We plot the error rate for plausible collusions with 2 up to 20 colluders. Code length and cutoff parameter for the continuous arcsine distribution are chosen according to the optimized parameter selection for the symmetric Tardos Codes in [21] with bound on the false negative error $\varepsilon_2 = 0.5$ and bound on the false positive error $n\varepsilon_1 = 0.1$ for $n = 1,000$ users. For each collusion size and each attack strategy we generated 100,000 fingerprint matrices and subsequently traced the colluders, except for the combinatorial attack for which we generated only 20,000 attacks for collusion sizes 12 to 20 due to their increasing complexity.

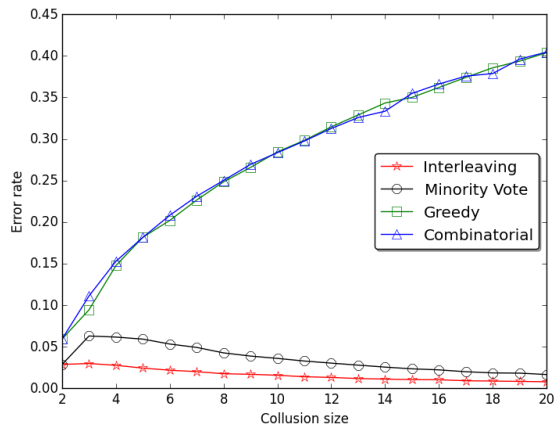

Figure 1. *'Detect one' error for $c = 2,\ldots,20$ colluders*

For $c = 20$, the stateful attack strategies effectuate that the resulting manipulated fingerprint y leads to more than 20 or 50 times higher error rates compared to the minority vote or interleaving attack, respectively. This being only an implication of the big picture, the results show that for the stateful strategies the error rate increases with growing collusion size (with parameters

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

adjusted accordingly), while it decreases or approximately stays the same for the stateless strategies.

Note that the code length was adapted according to the collusion size. Therefore, the difference between stateless and stateful strategies indeed increases with the collusion size, indicating that large collusions can profit a lot more from using a stateful attack strategy than small collusions. This is expected as the colluders get a more precise estimate of the secret bias vector p and consequently the estimates of their own scores improve.

Moreover we see the minority vote attack considerably exceeding the interleaving attack, although the difference shrinks with growing collusion size. This observation is consistent with our other setups in which the minority vote attack outperforms the interleaving attack.

Results for various decoders: In order to examine the alteration of errors in more detail and for various parameter settings we provide error rates for stateful and stateless attacks when traced by the tracing-decoders suggested in literature that were introduced in section 2, which are denoted here as 'S', 'L', 'O' and 'MF' for Škorić, Laarhoven, Oosterwijk and Meerwald-Furon, respectively. In addition to Setups A and B used in [8], we provide results for a third setting that is more favorable to the pirates and, more importantly, features a larger collusion size. The exact parameters are chosen to be:

Setup A: $m = 256$ $c = 3$ $c_0 = 6$ $t = 5.5 * 10^{-4}$

Setup B: $m = 1024$ $c = 6$ $c_0 = 10$ $t = 3.3 * 10^{-4}$

Setup C: $m = 2048$ $c = 15$ $c_0 = 15$ $t = 1.1 * 10^{-4}$

Following the notation from section 2, m denotes the code length, c stands for the actual collusion size, c_0 is the selected maximum collusion size to be resistant against and t represents the cutoff. In all three setups we generate fingerprints for $n = 1,000$ users and perform 10,000 attacks for each collusion strategy. We denote these strategies as follows:

- IL: Interleaving attack
- MIN: Minority Vote attack
- GRE_S: Greedy attack with Škorić collusion-decoder
- GRE_L: Greedy attack with Laarhoven collusion-decoder
- GRE_O: Greedy attack with Oosterwijk collusion-decoder
- COM_S: Combinatorial attack with Škorić collusion-decoder

As the above list conveys we try to adapt the greedy attack to accommodate the possibility of being scored by various tracing-decoders by letting the colluders compute their expected scores based on the score functions suggested by Škorić et al. [36], Oosterwijk et al. [31] and Laarhoven [20]. The resulting attacks are denoted with subscript 'S', 'L' and 'O', respectively. Note that for the combinatorial attack the symbols in positions not covered by the algorithm itself are decided upon by the greedy attack expecting the symmetric score function of the Škorić decoder. Here we did not include other collusion-decoders as at most $2^{c-1}-1$ such positions can exist, thus making almost no difference in Setups A and B.

The results in Table 1 show that for Setup A stateful attacks strikingly improve the colluders chances against the Škorić tracing-decoder and increase them facing the Meerwald-Furon tracing-decoder, e.g. opposing the Škorić tracing-decoder for the stateful attacks featuring the Škorić collusion-decoder we obtain 835 and 935 errors, respectively, in comparison to only 442 errors for the minority vote attack. Yet for both Laarhoven and Oosterwijk tracing-decoder stateful attacks fail to strengthen the most impactful stateless attack, i.e. the minority vote attack. For every tested decoder the combinatorial attack delivers the highest error rate among the stateful attacks.

	Tracing-decoder			
Attack	S	L	O	MF
IL	280	23	130	32
MIN	442	299	389	77
GRE_S	835	67	165	75
GRE_L	752	110	199	124
GRE_O	672	66	132	48
COM_S	935	139	268	134

Table 1: Absolute number of errors at 10,000 attempts in Setup A with $m = 256$, $c = 3$, $c_0 = 6$, $t = 5.5 * 10^{-4}$

	Tracing-decoder			
Attack	S	L	O	MF
IL	75	0	31	2
MIN	91	29	36	0
GRE_S	626	20	71	6
GRE_L	309	17	44	7
GRE_O	236	2	20	2
COM_S	597	39	82	5

Table 2: Absolute number of errors at 10,000 attempts in Setup B with $m = 1024$, $c = 6$, $c_0 = 10$, $t = 3.3 * 10^{-4}$

In Table 2 the errors affected by the different attacks are listed for Setup B. Here we observe an overall improvement of the stateful attacks' stance compared to Setup A. The combinatorial attack consistently outperforms stateless attacks, while the greedy attack only struggles versus the Laarhoven tracing-decoder. Notably the greedy attack with Oosterwijk collusion-decoder performs worst among the greedy attacks.

	Tracing-decoder			
Attack	S	L	O	MF
IL	1851	193	619	201
MIN	2186	528	135	0
GRE_S	8755	1944	456	106
GRE_L	5882	2685	1404	2728
GRE_O	5553	2207	1455	2098
COM_S	8817	2138	515	118

Table 3: Absolute number of errors at 10,000 attempts in Setup C with $m = 2048$, $c = 15$, $c_0 = 15$, $t = 1.1 * 10^{-4}$

Table 3 finally presents Setup C in which the stateful attacks generally outmatch stateless ones. Even the Laarhoven tracing-decoder that successfully limited stateful efforts in Setups A and B is now overcome. Note however that the greedy attack with Škorić collusion-decoder and the combinatorial attack are exceptions to

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

stateful attacks producing high error rates when facing the Oosterwijk or Meerwald-Furon tracing-decoder.

All in all we confirm again that with growing collusion size the attack strategies that operate location-dependent do have an edge compared to those using a memoryless collusion channel. Overall the greedy attack with Laarhoven collusion-decoder seems to consistently produce many errors across all tracing-decoders and setups. The greedy attack with Škorić collusion-decoder excels when actually being scored by the Škorić tracing-decoder, accounts for decently high error rates for most other setups and tracing-decoders, while performing below average for some, most notably in Setup C versus Oosterwijk and Meerwald-Furon tracing-decoders as mentioned above. Lastly, Oosterwijk collusion-decoder does not seem to yield results favorable to the colluders as it produces the least number of errors among all greedy attacks in both Setup A and Setup B. The only setting in which it barely exceeds the Laarhoven collusion-decoder is in Setup C versus the Oosterwijk tracing-decoder. From the collusion's point of view the combinatorial attack broadly performs well, often being the best stateful attack and almost never lagging behind by far if not, except in Setup C opposing Oosterwijk and Meerwald-Furon tracing-decoders. However, as mentioned in section 3, it is impractical due to its need to distinguish symbols.

4.2 Evaluation in a 'detect many' scenario

The 'detect many' scenario is the more common and realistic of both scenarii. Distributors would want to find preferably every pirate to shut down future malicious actions. As mentioned earlier thresholding becomes an interesting challenge. First of all we tested as to what happens in case a static threshold is used. For this purpose we track error rates across a broad range of thresholds. Then we employ techniques from rare event analysis [12] in order to illustrate how our stateful attacks work and highlight their use cases from the colluders' point of view. Lastly we view detailed accusation rates for the three setups introduced in 4.1.

Results for the Škorić decoder: In the following we provide two plots comparing false negative error rates with tracing- and collusion-decoder based on the Škorić decoder. The first tracks the behavior of a small collusion ($c = 5$), while we examine a large collusion ($c = 25$) in our second setup. Code length is $m = 1,614$ and $m = 31,485$, respectively, chosen together with the cutoff parameter according to the improved parameter selection proposed in [21] with $\varepsilon_1 = 0.001$. 100,000 attacks have been simulated for each collusion channel, except the combinatorial attack with 25 colluders, for which we depict the results of only 2,000 attacks due to its computational complexity. But again, these 2,000 attacks suggest that greedy and combinatorial attack lead to similar error rates. We depict thresholds that show the range of error rates encountered for all attack strategies.

Note that throughout this work we use comparably high values of ε_1. However, a lower value of ε_1 leads to a higher threshold. In turn this results in even more false negative errors affected by the stateful attacks as mirrored by Figure 2 and Figure 3. Therefore we can get rid of some computational complexity without invalidating the evidence we gather.

Showcased in Figure 2 we can demonstrate that the errors of the tracing algorithm affected by greedy and combinatorial attack are comparably close. Moreover for a fixed threshold they both result in error rates higher than for the minority vote attack by a factor ranging from about 3 to 5 for error rates below 10% that vanish when the error probability approaches 1. Together with the results presented in Figure 3 it validates what we concluded from our findings shown in Figure 1, that is the difference between stateful and stateless attacks only grows in c: For 25 colluders we observe huge differences in the error rate, e.g. for a threshold $Z = 900$ we have false negative error rates of $1 * 10^{-5}, 3 * 10^{-4}$ and $4 * 10^{-1}$ for interleaving, minority vote and greedy attack, respectively. That is, the greedy attack can affect the tracing algorithm to produce error rates more than three magnitudes higher than the minority vote attack for static thresholds.

We now turn our attention towards the question how stateful attacks impact the distribution of scores in addition to why and when colluders could improve by applying them.

In Figures 4 and 5 we depict score distributions in Setup A and Setup C for interleaving and greedy attack. They suggest that stateful attacks (the combinatorial attack leads to similar distributions) do not influence scores of innocent users significantly. The fact that thresholding with rare event analysis relies only on the distribution of scores of innocents causes the inability of such dy-

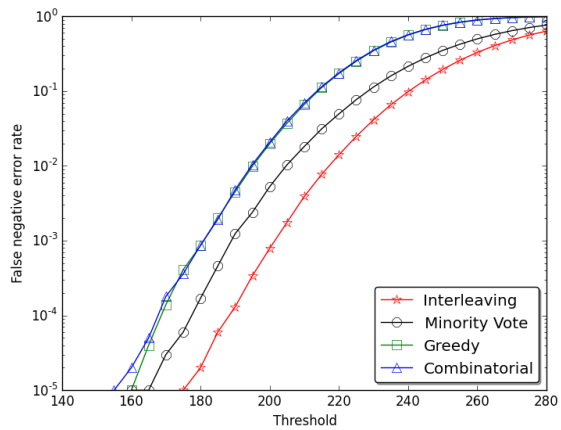

Figure 2. *'Detect many' false negative error for $c = c_0 = 5$ and $m = 1614$*

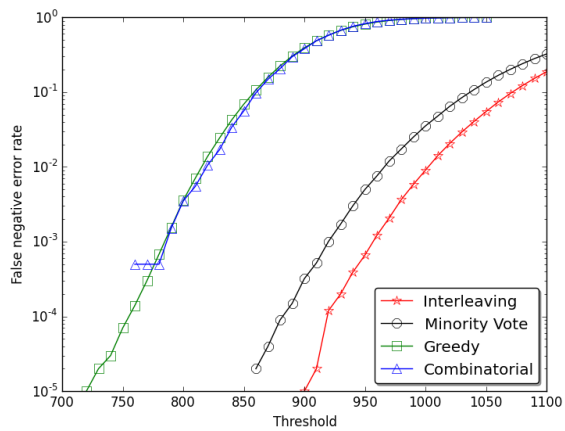

Figure 3. *'Detect many' false negative error for $c = c_0 = 25$ and $m = 31485$*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

namic threshold estimators – let alone static thresholds – to adapt our stateful attacks.

Depicted in the figures, for each run we estimated a threshold with bound on the false positives $\varepsilon_1 = 0.001$. Subsequently we averaged over all these thresholds with averages for interleaving and greedy attack differing by less than 0.015 in Figure 4 and by less than 1.4 in Figure 5. This experimentally validates our claim that dynamic thresholds cannot properly adapt to the change of colluder scores exhibited by our stateful attacks.

Moreover in Figure 4 and Figure 5 we observe what causes the increase in error rates. The colluders' scores vary less on an equally high level, or more technically, they are still Gaussian-distributed with the same mean but smaller variance. Consequently, we obtain accusation rates that reinforce stateless deviations away from 50%. In other words, if for a stateless attack in the same setting more than 50% of colluders could be accused, then it will be even more in case they use a stateful attack. But if less than 50% are accused in the stateless case, then it will be even less for a stateful attack. The former can be observed in Figure 4, where in Setup A this corresponds to 86% of pirates being accused instead of 76%, while the latter shows in Figure 5 with less

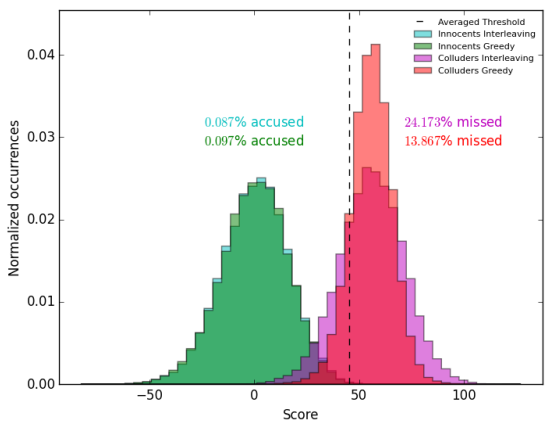

Figure 4. *Distribution of scores normalized over 10,000 attempts in Setup A*

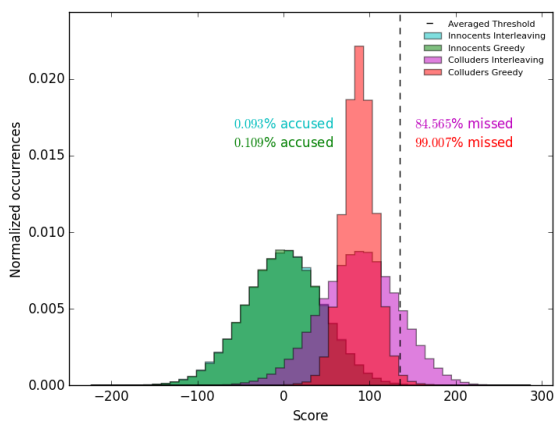

Figure 5. *Distribution of scores normalized over 10,000 attempts in Setup C*

than 1% of the pirates being caught instead of more than 15%.

Depending on the pirates' goals this behavior may be advantageous and highly desired. Not only seems it unrealistic that pirates redistribute a forged copy if they assume the majority of them to be caught anyways, but in case they did, then they might use a scapegoat strategy to begin with. Otherwise they try to avoid many – maybe even a single – of the colluders from being accused, which is easier if the highest scores are lower, as is the case for our stateful attacks. In addition, dynamic or iterative schemes hold inherent the possibility to iteratively uncover many perpetrators from one accused pirate in the first place, further strengthening the claim that pirates would want to avoid even a few of their own from being detected. Hence, we assume that collusions assume that less than 50% – preferably none – of their members are being accused in the first place. In that case our stateless attacks promise an advantage to the attackers.

Results for various decoders: To back up these considerations of when stateful attacks are reasonably applicable and of how large an advantage they can yield, we provide the percentages of users wrongly accused ('WA'), that is innocent users exceeding the threshold, and of colluders who escaped detection ('ED'), that is those pirates acquitted due to their scores being below the threshold. For all three setups we give results for 1,000 runs with a threshold chosen according to rare event analysis with $\varepsilon_1 = 0.001$. Additionally, for Setup A we give absolute false positive and false negative occurrences to qualify the statement emerging from the percentage of escaped colluders.

Attack	Tracing-decoder							
	S		L		O		MF	
	WA	ED	WA	ED	WA	ED	WA	ED
IL	0.1	25.4	0.1	10.9	0.1	31.6	0.1	12.3
MIN	0.1	27.4	0.1	31.6	0.1	47	0.1	20.9
GRE_S	0.1	14.5	0.1	4.1	0.1	27.8	0.1	4.7
GRE_L	0.1	14.5	0.1	5.7	0.1	32.9	0.1	5.9
GRE_O	0.1	15.7	0.1	1.9	0.1	24.2	0.1	2.3
COM_S	0.1	12.8	0.1	3.8	0.1	28.9	0.1	4.4

Table 4: Percentages of wrongly accused users and escaped colluders at 1,000 attempts in Setup A with $m = 256$, $c = 3$, $c_0 = 6$, $t = 5.5 * 10^{-4}$

The results in Table 4 show that as long as the tracing-decoder is working properly stateful attacks do not help the colluders, as can be seen for all tracing-decoders. We observe the percentages of colluders who escaped detection being higher for stateless attacks. Again, considering the change in score distribution this is expected, because more colluders are caught in the stateful attack if the threshold is lower than the colluders' mean (see Figure 4). Furthermore we also observe that the difference varies depending on the tracing-decoder, with Škorić and Oosterwijk decoders discriminating much less than Laarhoven and Meerwald-Furon decoders.

These results seem to imply that location-dependent attacks perform strictly worse in Setup A. However, focusing solely on the number of occurrences of a false negative error, we see below in Table 5 that clearly for the Škorić tracing-decoder and barely for the Meerwald-Furon tracing-decoder stateful attacks do yield

DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

an advantage for the colluders in Setup A already. This observation is consistent with the 'detect one' results in Table 1.

Note that the high number of false positive errors is due to $\varepsilon_1 = 0.001$. In expectation we accuse 1 of 1,000 innocent users, with 997 innocents in each run. The fact that much less than 1,000 false positive errors occur is due to multiple wrongly accused users per run still accumulate to only one false positive error.

Attack	Tracing-decoder							
	S		L		O		MF	
	FP	FN	FP	FN	FP	FN	FP	FN
IL	648	11	644	0	631	1	621	1
MIN	630	15	656	8	642	9	638	1
GRE_S	658	37	631	0	642	0	640	0
GRE_L	639	20	615	0	662	3	633	0
GRE_O	634	31	655	0	628	0	660	1
COM_S	597	33	639	2	647	2	642	2

Table 5: Absolute number of false positive and false negative errors at 1,000 attempts in Setup A with $m = 256$, $c = 3$, $c_0 = 6$, $t = 5.5 * 10^{-4}$

Attack	Tracing-decoder							
	S		L		O		MF	
	WA	ED	WA	ED	WA	ED	WA	ED
IL	0.1	28.5	0.1	8.4	0.1	34.9	0.1	8.7
MIN	0.1	32.7	0.1	31.8	0.1	44.7	0.1	0
GRE_S	0.1	16.7	0.1	12.8	0.1	48.6	0.1	7.8
GRE_L	0.1	18.6	0.1	0.8	0.1	33.8	0.1	1.4
GRE_O	0.1	20.2	0.1	0.1	0.1	18.7	0.1	0.1
COM_S	0.1	16.2	0.1	13.8	0.1	47.7	0.1	7.8

Table 6: Percentages of wrongly accused users and escaped colluders at 1,000 attempts in Setup B with $m = 1024$, $c = 6$, $c_0 = 10$, $t = 3.3 * 10^{-4}$

Setup B has a similar trend, as shown in Table 6. However, if the collusion uses the Škorić collusion-decoder the huge discrimination of the Meerwald-Furon tracing decoder vanishes and the escape percentages of stateless attacks are exceeded.

Moreover we observe that astoundingly Laarhoven and Oosterwijk collusion-decoders seem to fail miserably for the colluders when traced by the Laarhoven or Meerwald-Furon tracing-decoder. This is contrary to Setup A, where the Laarhoven collusion-decoder proves to be the best choice except when scored by the symmetric score function of the Škorić decoder.

Another interesting fact is that correctly anticipating the Škorić tracing-decoder does actually lessen the fraction of missed colluders. When looking at the absolute number of false negative errors (GRE_S : 10, GRE_L : 4, GRE_O : 2, see Table 7) we see this turned upside down. This is due to the fact that expecting the tracing-decoders actually employed reduces the variance the most, leading to higher accusation rates in unfavorable settings such as Setup A and B.

The combinatorial attack delivers notably strong results among the stateful attacks, together with GRE_S even exceeding the amount of colluders missed by the Oosterwijk tracing-decoder in case of stateless attacks.

Apart from the afore-mentioned results of the Škorić tracing-decoder Table 7 gives no further insight as all other tracing-decoders always accuse at least one colluder, but is nonetheless displayed for the sake of completeness.

Attack	Tracing-decoder							
	S		L		O		MF	
	FP	FN	FP	FN	FP	FN	FP	FN
IL	642	0	609	0	615	0	643	0
MIN	650	1	610	0	648	0	650	0
GRE_S	661	10	629	0	633	0	633	0
GRE_L	661	4	633	0	623	0	619	0
GRE_O	600	2	636	0	637	0	668	0
COM_S	626	11	654	0	627	0	656	0

Table 7: Absolute number of false positive and false negative errors at 1,000 attempts in Setup B with $m = 1024$, $c = 6$, $c_0 = 10$, $t = 3.3 * 10^{-4}$

Attack	Tracing-decoder							
	S		L		O		MF	
	WA	ED	WA	ED	WA	ED	WA	ED
IL	0.1	84.4	0.1	63.8	0.1	79.6	0.1	64.3
MIN	0.1	86,8	0.1	76.0	0.1	72.6	0.1	0
GRE_S	0.1	99.0	0.1	84.0	0.1	83.1	0.1	57.8
GRE_L	0.1	94.7	0.1	79.5	0.1	88.2	0.1	79.4
GRE_O	0.1	93.5	0.1	68.4	0.1	82.0	0.1	67.5
COM_S	0.1	99.0	0.1	84.1	0.1	83.0	0.1	57.5

Table 8: Percentages of wrongly accused users and escaped colluders at 1,000 attempts in Setup C with $m = 2048$, $c = 15$, $c_0 = 15$, $t = 1.1 * 10^{-4}$

In contrast to Setup A and Setup B, in Setup C we see significant increases in the error percentages against all tracing-decoders as Table 8 illustrates. Of course this is partly due to the setup favoring the colluders based on the small code length compared to the collusion size. On the other hand stateful attacks obviously surpass both interleaving and minority vote attack for all but one decoder.

Note that Setup C should even favor the tracing-decoders of Laarhoven and Meerwald and Furon, because the guessed number of colluders is precise. In fact both have on average the lowest error rates, with the difference being a lot more significant in case of Meerwald-Furon.

While the greedy attack with Oosterwijk collusion-decoder struggles when faced with the Laarhoven tracing-decoder and both combinatorial attack and greedy attack with Škorić collusion-decoder do not quite meet the interleaving attack's percentage of missed colluders opposing the tracing-decoder of Meerwald and Furon, the Laarhoven collusion-decoder guarantees increased error percentages regardless of the actual tracing-decoder.

Finally Table 9 displays the absolute number of false positive and false negative errors in Setup C. We observe the false negative error occurrences of stateful attacks towering above those of both

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.8.MWSF-085

Attack	Tracing-decoder							
	S		L		O		MF	
	FP	FN	FP	FN	FP	FN	FP	FN
IL	615	77	650	0	653	4	639	1
MIN	654	126	614	9	658	1	615	0
GRE_S	634	858	641	91	648	2	632	1
GRE_L	638	479	634	118	637	36	656	137
GRE_O	625	422	645	62	607	24	637	50
COM_S	637	881	648	98	638	2	628	0

Table 9: Absolute number of false positive and false negative errors at 1,000 attempts in Setup C with $m = 2048$, $c = 15$, $c_0 = 15$, $t = 1.1 * 10^{-4}$

minority vote and interleaving attack when facing the Škorić and Laarhoven tracing-decoders. For the Oosterwijk and Meerwald-Furon tracing-decoders the greedy attacks with Laarhoven and Oosterwijk collusion-decoders surpass any other attack strategy by a long shot.

Summing up on the 'detect many' results, we note that the Škorić collusion-decoder delivers more consistent clearing percentages than both Laarhoven and Oosterwijk collusion-decoders as opposed to the 'detect one' scenario and the absolute false negative occurrences in which the Laarhoven collusion-decoder proved to be the most favorable to the collusion. Nevertheless each choice of a collusion-decoder has its own set of strengths and weaknesses depending on both the setting and the tracing-decoder. Setup A is punishing the collusion for applying stateful attacks, with differences reduced in Setup B. Setup C turns this around, rewarding colluders who deviate from the location-independence and make use of the stateful attacks.

5 Conclusion and Future Work

Most research regarding collusion secure fingerprinting codes focuses on optimal fingerprint matrix generation or efficient tracing algorithms. Though there exist prior publications that present or discuss sophisticated collusion attacks (e.g. [26], [11], [18], [15]), to the best of our knowledge, the class of stateful attacks has not been considered before in the field of modern fingerprinting codes such as the Tardos codes [35]. We defined the class of stateful attacks as attacks for which the colluders need not select the symbol of the current position of the fingerprint independently from the (symbols of) other positions. This enables new possibiities for the colluders that have to be considered by the code designers and their decoders as well. To prove this, we introduced two stateful attack strategies denoted as greedy attack and combinatorial attack that show competitive behavior to those attacks discussed in literature. If the colluders intend to escape the accusation of the tracing algorithm for each of them, they might benefit from applying a stateful attack strategy. We compared the stateful attacks introduced in this work to the most poweful stateless attacks commonly discussed in literature and showed that these new attacks pose new possibilities for significant improvement for the colluders. However, some fingerprinting parameters are crucial for successful detection or tracing. If the number of expected colluders c_0 is significantly larger than the actual collusion size c, stateful attacks are likely to help the tracers before the colluders. Also, the colluders have to guess what decoder the tracers might apply for tracing them, in order to conduct a stateful attack. A disadvantageously selected decoder for creating the attack might return in a higher chance to get caught by the tracers.

Altogether stateful attacks save more colluders only when used in the right scenario. That is, if the setting is punishing the colluders a lot in the first place, then both greedy and combinatorial attack are likely to be counterproductive. Applied in the right setting though, they yield advantages over conventional stateless attacks. As previously reasoned, we think that the pirates may usually (at least think to) be in such a favorable scenario if they decide to collude. Hence, stateful attacks provide a set of new and strong tools to the party of pirates.

However, many things such as the diverse results for the different colluder- and tracing-decoder are not yet full understood. For this reason, we plan to extend the testset to other decoders, e.g. [8], and methods to calculate the accusation threshold, e.g. [33], in order to get more insights of this new class of attacks. Beside the empirical evaluations, the theoretic background to this attacks is still void. While for collusions of two an analytical proof of the superiority of stateful attacks over stateless attacks is straight forward, proofs for larger collusions promise to be tedious. Future elaborations should bring more light into the class of stateless attacks.

References

[1] Ehsan Amiri and Gábor Tardos. High rate fingerprinting codes and the fingerprinting capacity. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 336–345, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[2] Waldemar Berchtold and Marcel Schäfer. Performance and code length optimization of joint decoding Tardos fingerprinting. In *Proceedings of the on Multimedia and security*, MM&Sec '12, pages 27–32, New York, NY, USA, 2012. ACM.

[3] Oded Blayer and Tamir Tassa. Improved versions of Tardos' fingerprinting scheme. *Designs, Codes and Cryptography*, 48:79–103, 2008. 10.1007/s10623-008-9200-z.

[4] Dion Boesten and Boris Škorić. Asymptotic fingerprinting capacity for non-binary alphabets. In Tom Filler, Tom Pevn, Scott Craver, and Andrew Ker, editors, *Information Hiding*, volume 6958 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2011.

[5] Dion Boesten and Boris Škorić. Asymptotic fingerprinting capacity in the combined digit model. In Matthias Kirchner and Dipak Ghosal, editors, *Information Hiding*, volume 7692 of *Lecture Notes in Computer Science*, pages 255–268. Springer Berlin Heidelberg, 2013.

[6] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, Sept. 1998.

[7] Frédéric Cérou, Teddy Furon, and Arnaud Guyader. Experimental assessment of the reliability for watermarking and fingerprinting schemes. Research Report RR-6636, INRIA, 2008.

[8] Teddy Furon and Mathieu Desoubeaux. Tardos codes for real. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, pages 24–29, Dec 2014.

[9] Teddy Furon, Arnaud Guyader, and Frédéric Cérou. On the design and optimization of tardos probabilistic fingerprinting codes. In Kaushal Solanki, Kenneth Sullivan, and Upamanyu Madhow, ed-

itors, *Information Hiding*, volume 5284 of *Lecture Notes in Computer Science*, pages 341–356. Springer Berlin Heidelberg, 2008.

[10] Teddy Furon and Luis Pérez-Freire. EM Decoding of Tardos Traitor Tracing Codes. In *ACM Multimedia and Security*, Princeton, United States, September 2009.

[11] Teddy Furon and Luis Pérez-Freire. Worst case attacks against binary probabilistic traitor tracing codes. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 56 –60, dec. 2009.

[12] Arnaud Guyader, Nicolas Hengartner, and Eric Matzner-Løber. Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics and Optimization*, 2011.

[13] Yen-Wei Huang and Pierre Moulin. Saddle-point solution of the fingerprinting capacity game under the marking assumption. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2256 –2260, 28 2009-july 3 2009.

[14] Yen-Wei Huang and Pierre Moulin. On fingerprinting capacity games for arbitrary alphabets and their asymptotics. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2571–2575, July 2012.

[15] Yen-Wei Huang and Pierre 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.

[16] Sarah Ibrahimi, Boris Škorić, and Jan-Jaap Oosterwijk. Riding the saddle point: asymptotics of the capacity-achieving simple decoder for bias-based traitor tracing. *EURASIP Journal on Information Security*, 2014(1), 2014.

[17] Minoru Kuribayashi. Experimental assessment of probabilistic fingerprinting codes over awgn channel. In Isao Echizen, Noboru Kunihiro, and Ryoichi Sasaki, editors, *Advances in Information and Computer Security*, volume 6434 of *Lecture Notes in Computer Science*, pages 117–132. Springer Berlin Heidelberg, 2010.

[18] Minoru Kuribayashi. Tardos's fingerprinting code over AWGN channel. In Rainer Böhme, Philip W.L. Fong, and Reihaneh Safavi-Naini, editors, *Information Hiding*, volume 6387 of *Lecture Notes in Computer Science*, pages 103–117. Springer Berlin Heidelberg, 2010.

[19] Minoru Kuribayashi. A new soft decision tracing algorithm for binary fingerprinting codes. In *Advances in Information and Computer Security - 6th International Workshop, IWSEC 2011, Tokyo, Japan, November 8-10, 2011. Proceedings*, pages 1–15, 2011.

[20] Thijs Laarhoven. Capacities and capacity-achieving decoders for various fingerprinting games. In *Proceedings of the 2Nd ACM Workshop on Information Hiding and Multimedia Security*, IHMMSec '14, pages 123–134, New York, NY, USA, 2014. ACM.

[21] Thijs Laarhoven and Benne Weger. Optimal symmetric tardos traitor tracing schemes. *Designs, Codes and Cryptography*, 71(1):83–103, 2014.

[22] Peter Meerwald and Teddy Furon. Iterative single tardos decoder with controlled probability of false positive. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1 –6, july 2011.

[23] Peter Meerwald and Teddy Furon. Towards joint tardos decoding: The 'don quixote' algorithm. In *Information Hiding - 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers*, pages 28–42, 2011.

[24] Peter Meerwald and Teddy Furon. Towards practical joint decoding of binary Tardos fingerprinting codes. *IEEE Transactions on Information Forensics and Security*, 7(4):1168–1180, April 2012.

[25] Pierre Moulin. Universal fingerprinting: Capacity and random-coding exponents. *CoRR*, abs/0801.3837, 2008.

[26] Pierre Moulin and Negar Kiyavash. Performance of random fingerprinting codes under arbitrary nonlinear attacks. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–157–II–160, April 2007.

[27] Pierre Moulin and Joseph A. O'Sullivan. Information-theoretic analysis of information hiding. *Information Theory, IEEE Transactions on*, 49(3):563 – 593, mar 2003.

[28] Koji Nuida, Satoshi Fujitsu, Manabu Hagiwara, Takashi Kitagawa, Hajime Watanabe, Kazuto Ogawa, and Hideki Imai. An improvement of discrete tardos fingerprinting codes. *Des. Codes Cryptography*, 52:339–362, September 2009.

[29] Koji Nuida, Manabu Hagiwara, Hajime Watanabe, and Hideki Imai. Optimization of tardos's fingerprinting codes in a viewpoint of memory amount. In Teddy Furon, François Cayre, Gwenaël Doërr, and Patrick Bas, editors, *Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 279–293. Springer Berlin Heidelberg, 2007.

[30] Jan-Jaap Oosterwijk, Boris Škorić, and Jeroen Doumen. A capacity-achieving simple decoder for bias-based traitor tracing schemes. *IACR Cryptology ePrint Archive*, 2013:389, 2013.

[31] Jan-Jaap Oosterwijk, Boris Škorić, and Jeroen Doumen. Optimal suspicion functions for tardos traitor tracing schemes. In *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '13, pages 19–28, New York, NY, USA, 2013. ACM.

[32] Luis Pérez-Freire and Teddy Furon. Blind decoder for binary probabilistic traitor tracing codes. In *Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on*, pages 46–50, 2009.

[33] Marcel Schäfer, Sebastian Mair, Waldemar Berchtold, and Martin Steinebach. Universal Threshold Calculation for Fingerprinting Decoders using Mixture Models. In *Proceedings of the third ACM workshop on Information Hiding and Multimedia Security (IH&MMSEC 2015), June 17-19, 2015 Portland, OR, USA*, IH&MMSec '15, New York, NY, USA, June 2015. ACM.

[34] Antonino Simone and Boris Škorić. Accusation probabilities in tardos codes: beyond the gaussian approximation. *Des. Codes Cryptography*, 63(3):379–412, 2012.

[35] Gábor Tardos. Optimal probabilistic fingerprinting codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC), pp. 116-125*, 2003.

[36] Boris Škorić, Stefan Katzenbeisser, and Mehmet Celik. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. *Designs, Codes and Cryptography*, 46:137–166, 2008.

[37] Boris Skoric Škorić, Tatiana U. Vladimirova, Mehmet Celik, and Joop C. Talstra. Tardos Fingerprinting is Better Than We Thought. *Information Theory, IEEE Transactions on*, 54(8):3663–3676, Aug. 2008.

[38] Ying Wang and Pierre Moulin. Capacity and optimal collusion attack channels for gaussian fingerprinting games. In Edward J. III Delp and Ping Wah Wong, editors, *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX, 65050J*, volume 6505, pages 65050J–65050J–9, 2007.