Tardos’ s Fingerprinting Code over AWGN Channel

Kuribayashi, Minoru

Tardos's Fingerprinting Code over AWGN Channel

Minoru Kuribayashi[1]

Graduate School of Engineering, Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe, Hyogo, 657-8501 Japan.
kminoru@kobe-u.ac.jp

Abstract. Binary fingerprinting codes with a length of theoretically minimum order were proposed by Tardos, and the traceability has been estimated under the well-known marking assumption. In this paper, we estimate the traceability and the false-positive probability of the fingerprinting code over AWGN channel, and propose a new accusation algorithm to catch more colluders with less innocent users. The design of our algorithm is based on the symmetric accusation algorithm proposed by Škorić et al. that focuses on the characteristic of the p.d.f. of the correlation scores. The proposed algorithm first estimates the strength of noise added to the code, and then calculates the specific correlation scores among candidate codewords using the characteristic of the noisy channel. The scores are finally classified into guilty and innocent by the threshold obtained from the p.d.f. The performance of the proposed tracing algorithm is evaluated by Monte Carlo simulation.

1 Introduction

Digital fingerprinting is used to trace the illegal users, where a unique ID known as a digital fingerprint [10] is embedded into the content before distribution. When a suspicious copy is found, the owner can identify illegal users by extracting the fingerprint. Since each user purchases contents involving his own fingerprint, the fingerprinted copy slightly differs with each other. Therefore, a coalition of users will combine their different marked copies of the same content for the purpose of removing/changing the original fingerprint. One of the solution is to encode the fingerprint information by a binary code, known as collusion secure code.

An early work on designing collusion-resistant binary fingerprinting codes was presented by Boneh and Shaw [1] underlying the principle referred to as the *marking assumption.* In this case, a fingerprint is a set of redundant digits which are distributed in some random positions of an original content. When a coalition of users attempts to discover some of the fingerprint positions by comparing their marked copies for differences, the coalition may modify only those positions where they find a difference in their fingerprinted copies. A c-secure code guarantees the tolerance for the collusion attack with c pirates or less. Tardos [9] has proposed a probabilistic c-secure code with error probability

ϵ which has a length of theoretically minimal order with respect to the number of colluders. Many researchers have focused on the characteristics of the code to reduce the code length under the marking assumption. One of the interesting reports is presented by Škorić et al. [2] about the symmetric version of the tracing algorithm and Gaussian approximation. Based on the report, the code length is further shortened under a fixed false-positive probability.

Considering about the realistic situation, a fingerprinting codeword is embedded into digital contents using a watermarking technique. Because of the characteristic of the watermark extraction, the codeword is distorted by signal processing operations as well as the collusion attack. Nuida et al. [7] gave a security proof under an assumption weaker than the marking assumption. The code length was evaluated under the binary symmetric channel with a certain error rate. Once a code length is fixed in an application, however, the important factor is how to detect as many colluders as possible with small and constant false-positive probability.

In this paper, we study the tracing algorithm of Tardos's fingerprinting code under the following two assumptions. One is that a codeword is modulated by BPSK at embedding into digital contents. The other is that a pirated codeword produced by collusion attack is further distorted by transmitting over AWGN channel. Different from the conventional assumption that allows bit flips of the pirated codeword, the addition of white Gaussian noise is more realistic even if the robust watermarking method is applied to embed the fingerprint into digital contents. We first attempt to detect colluders directly from the degraded codeword using soft decision method similar to the detection procedure of error correcting codes. Then, we further reduce the probability of false-positive by classifying the elements of the distorted codeword into reliable ones and the others, and detect colluders with two steps. The first step reduces the candidates of suspicious users using only reliable elements, and the second step further narrows down the suspicious users using the whole codeword by the properly designed threshold which is calculated under the Gaussian assumption of the correlation score. The proposed approach can reduce the probability of false-positive and can detect as many colluders as possible.

2 Preliminaries

2.1 Tardos Code

In this section, we first review the original Tardos's binary fingerprinting code [9]. Let N be the allowable number of users in a fingerprinting system. The Tardos fingerprinting scheme distributes a binary codeword of length L to each user. The codewords are arranged as an $N \times L$ matrix $\boldsymbol{X}$, where the j-th row corresponds to the fingerprint given to the j-th user. The generation of the matrix $\boldsymbol{X}$ is composed of two steps.

1. A distributor is supposed to choose the random variables $0 < p_i < 1$ independently for every $1 \leq i \leq L$, according to a given bias distribution $\boldsymbol{P}$, which satisfies the following conditions.

- $t = 1/300c$
- $0 < t' < \pi/4 \ , \ \sin^2 t' = t \ , \ r_i \in [t', \pi/2 - t']$
- $p_i = \sin^2 r_i \ , \ t \leq p_i \leq 1 - t$

Where r_i is uniformly and randomly selected from the above range.

2. Each entry $X_{j,i}$ of the matrix $\boldsymbol{X}$ is selected independently from the binary alphabet $\{0, 1\}$ with $\Pr(X_{j,i} = 1) = p_i$ and $\Pr(X_{j,i} = 0) = 1 - p_i$ for every $1 \leq j \leq N$.

Let $\mathcal{C}$ be a set of colluders and c be the number of colluders. Then we denote by $\boldsymbol{X_C}$ the $c \times L$ matrix of codewords assigned to the colluders. Depending on the attack strategy ρ, the fingerprint $\boldsymbol{y} = (y_1, \ldots, y_L), y_i \in \{0, 1\}$ contained in a pirated copy is denoted by $\boldsymbol{y} = \rho(\boldsymbol{X_C})$. In a tracing (accusation) algorithm $\mathcal{A}$, a correlation score S_j of the j-th user is calculated

$$S_j = \sum_{i=1}^{L} y_i U_{j,i} \ , \tag{1}$$

where

$$U_{j,i} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}} & (X_{j,i} = 1) \\ -\sqrt{\frac{p_i}{1-p_i}} & (X_{j,i} = 0) \end{cases} . \tag{2}$$

If S_j exceeds a threshold Z, the j-th user is decided as guilty. The algorithm $\mathcal{A}$ outputs a list of colluders.

Škorić et al. [2] proposed a symmetric version of the correlation score:

$$S_j = \sum_{i=1}^{L} (2y_i - 1) U_{j,i} \ . \tag{3}$$

The traceability are usually evaluated in terms of the probability ϵ_1 of accusing innocent users and the probability of missing all colluders ϵ. In order to guarantee that the probability of accusing innocent users is below ϵ_1, the inequality $L > 2\pi^2 c^2 \log(N/\epsilon_1)$ [2] must be satisfied. The number of traceable colluders depends on the design of threshold Z. Referring to the Central Limit Theorem (CLT) in [2] [3], the distribution of the score S_j for innocent users is approximated to be Gaussian distribution $N(0, L)$ because it is a sum of L elements in Eq.(3). Under the Gaussianity assumption, the probability of false-positive that the j-th innocent user is accused is represented as follows.

$$\Pr\left(S_j > Z, j \in \mathcal{I}\right) = \frac{\epsilon_1}{N} = \frac{1}{2} \mathrm{erfc}\left(\frac{Z}{\sqrt{2L}}\right) \ , \tag{4}$$

where $\mathcal{I}$ stands for a set of innocent users and erfc() is the complementary error function. Hence, the threshold Z is written by a given ϵ_1 [5]:

$$Z = \sqrt{2L} \cdot \mathrm{erfc}^{-1}\left(\frac{2\epsilon_1}{N}\right) \ . \tag{5}$$

The validity of such a threshold Z is not assured because the use of the CLT is not recommended in statistics (i.e. integral over the tail of a p.d.f.). Instead of the use of CLT, conventional schemes calculated it based on the Chernoff's bound, union bound, etc. sacrificing the tightness of the upper bound. In this paper, we evaluate the validity of the threshold Z in Eq.(5) using Monte Carlo simulation and address the insight for the recommendation of the use of CLT.

At the collusion attack, c colluders try to find the positions of the embedded codeword from differences of their copies, and then to modify bits of the codeword in these positions. This attack model is called marking assumption formulated as follows.

Let we say that position i is undetectable for colluders in $\mathcal{C}$ if the codewords assigned to c colluders in $\mathcal{C}$ match in i-th position. Then, $y_i = X_{j,i}$ for any $j \in \mathcal{C}$. Under the marking assumption, colluders have no information on the i-th position of innocent users if it is undetectable.

2.2 Relaxation of Marking Assumption

Suppose that a codeword of fingerprint codes is binary and each bit is embedded into one of the segments of digital content without overlapping using a robust watermarking scheme. It is possible for malicious users, called colluders, to compare their fingerprinted copies of the content with each other to find the differences. In the situation, the positions that the bit of their codewords is different are detectable. The marking assumption states that any bit within a detectable position can be selected or even erased, while any bit without the position will be left unchanged in the pirated codeword. A fingerprint code is called totally c-secure if at least one of the colluders is traceable under the marking assumption with the condition that the number of colluders is at most c. Boneh and Shaw, however, proved that when $c > 1$, totally c-secure codes do not exist if the marking assumption is satisfied [1]. Under the weaker condition that one of innocent users will be captured with a tiny probability ϵ, a c-secure code with ϵ-error was constructed. Since then, the study of c-secure code has been investigated under the marking assumption. Although the assumption is reasonable to evaluate the performance of fingerprint codes, there is a big gap from practical cases as follows. Even if a watermarking scheme offers a considerable level of robustness, it is still possible to erase/modify the embedded bits with a non-negligible probability due to the addition of noise to a pirated copy. Therefore, the bits at the undetectable positions as well as the detectable ones may be erased/modified by the attacks for the watermarked signal.

In order to cover more practical cases, various relaxation of marking assumption have been introduced and several c-secure codes under those assumptions, called robust c-secure codes, have been proposed in [7], [4], [8], [6]. Among those assumptions, there are two common conditions: At least one of the colluders is traceable and the number of colluders is at most c. Their goal is mainly to estimate a proper code length L to satisfy that the probability of accusing an innocent user is below ϵ_1, which is dependent on the number of flipped bits at the undetectable position. Although the study of such a code length is meaningful, there is still a difficulty to adapt the fingerprint codes in a system. When the number of colluders is more than c, the false probability may be increased. Even more, the dependency with the number of flipped bits lefts the uncertainty of the code length. Colluders can take a stronger attack strategy for the underlying watermarking scheme in order to affect more the embedded fingerprint code, which is an unavoidable feature of watermarking schemes.

From the different viewpoint, it is an interesting challenge to design a proper threshold Z for a given false probability ϵ_1 under a fixed code length. Based on the CLT, for a given false probability ϵ_1, the threshold Z is calculated by Eq.(5). The study in [5] shows the detectable number of colluders using such a threshold. So, it is also interesting to estimate how many colluders will be able to be caught by a tracing algorithm. To the best of our knowledge, no report about the detectable number of colluders has been presented under a relaxed version of the marking assumption.

Suppose that a fingerprint code is equipped in a fingerprinting system. Then, the code length must be determined under the considerations of system policy and attack strategies such as the number of colluders and the amount of noise. Here, our interest is how to design the good tracing algorithm that can detect more colluders and less innocent users no matter how many colluders get involved in to generate a pirated copy and no matter how much amount of noise is added to the copy.

3 Performance of Tracing Algorithm

In this section, we forget about the limitation of c-secure code such that the number of colluders is less than c. The performance of conventional tracing algorithm based on a threshold Z and its variant are evaluated for arbitrary number $\tilde{c}$ of colluders.

3.1 Channel Model

The conventional analysis considers the case that colluders change several symbols of a pirated codeword in an attempt to attack directly a pirated copy. Regretfully, the attack model is merely a bit flip. If each symbol of codeword is embedded into digital contents assisted by a watermarking technique, the extracted symbol from a pirated copy must contain noise caused by attacks intended to remove/modify the symbol. In such a case, it is a reasonable assumption that the symbol is disturbed by additive white Gaussian noise, namely the codeword is transmitted through AWGN channel.

Let $\boldsymbol{y} = (y_1, \ldots, y_L)$ be the fingerprint produced from $\tilde{c}$ colluders' codes under the marking assumption. Namely, the fingerprint is represented by a binary code. Here, we assume that a binary fingerprint code is embedded into digital contents after the BPSK (Binary Phase Shift Keying) modulation, hence, $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_L)$, where $\hat{y}_i \in \{-1, 1\}$. The codeword is transmitted through AWGN channel.

1. BPSK modulation
   In the tracing algorithm of Eq.(3), each symbol of the pirated codeword is modulated into two kinds of symbols $\{-1, 1\}$ to calculate the correlation score S_j. Since the modulation can be performed at the embedding, we assume that a binary fingerprint codeword is modulated by BPSK before embedding.
2. AWGN channel
   Even if a robust watermarking method is used to embed the binary fingerprint code into digital contents, it must be degraded by attacks. In our assumption the effects caused by attacks are modeled by additive white Gaussian noise, and the noise is added after collusion attack. The degraded fingerprint codeword is represented by

   $$\hat{\boldsymbol{y}}' = \hat{\boldsymbol{y}} + \boldsymbol{e} \ , \tag{6}$$

   where $\boldsymbol{e}$ is the additive white Gaussian noise.

3.2 Hard and Soft Decision

The signal extracted from a pirated copy is represented by analog value $\hat{\boldsymbol{y}}'$. At the tracing algorithm in Eq.(3), however, the codeword y_i of a pirated copy must be binary bit in $\{0, 1\}$. If it is not binary, the design of threshold Z in Eq.(5) is not valid. A simple solution is to quantize $\hat{y}'_i$ into "-1" if $\hat{y}'_i < 0$, otherwise "1". In this solution, an extracted signal is first quantized into digital value, and then the tracing algorithm is performed to identify the colluders. This solution is analogous to the hard decision (HD) method in error correcting code. Here, there is an interesting question whether a soft decision (SD) method is applicable to the tracing algorithm by adaptively designing a proper threshold or not. In general, the performance of SD method is much better than the HD method in error correcting code.

The design of threshold in Eq.(5) is based on the Gaussian approximation of the score S_j. Referring to the CLT, the variance of S_j is L, and hence, the proper threshold Z_{HD} is calculated by the Eq.(5).

$$Z_{HD} = \sqrt{2L} \cdot \mathrm{erfc}^{-1}\left(\frac{2\epsilon_1}{N}\right) \tag{7}$$

Since a pirated codeword is distorted by AWGN channel, the effect on S_j is also approximated to Gaussian. Hence, if the variance σ_{SD}^2 of S_j using the SD method is obtained, the proper threshold Z_{SD} can be designed using the same equation as the case of HD method:

$$Z_{SD} = \sqrt{2\sigma_{SD}^2} \cdot \mathrm{erfc}^{-1}\left(\frac{2\epsilon_1}{N}\right) \ . \tag{8}$$

Because of the randomness in the generation of codeword, the variance σ_{SD}^2 can be calculated as follows.

1. Generate N' fingerprint codewords $X_{j',i}$ for $j' \notin \{1, \ldots, N\}$.
2. Calculate the correlation scores S'_j.
3. Compute the variance of S'_j, and output it as σ^2_{SD}.

The generated N'codewords $X_{j',i}$ are statistically uncorrelated with the pirated codeword. If N' is sufficiently large, a proper variance can be obtained by the above procedure, and finally, a proper threshold Z_{SD} is derived.

3.3 Numerical Comparison

For the comparison of the performance of HD and SD methods, the number of detected colluders and false-positive probability is plotted in Fig.1 and Fig.2, respectively. The number of users is $N = 10^4$, the code length is $L = 10^4$, SNR is fixed by 8 [dB], and the number of trials for Monte Carlo simulation is 10^5 in this experiment. In addition, the range of bias distribution p_i is given by setting $t = 0.000167$ ($c = 20$). In the SD method, the number of codewords to calculate σ^2_{SD} is $N' = 10^3$. In this experiment, we check the validity of the use of CLT to set the thresholds Z_{HD} and Z_{SD} from the targeted false-positive probability point of view, which is designed by $\epsilon_1 = 10^{-4}$.

From the Fig.1, we can see that the HD method detects more colluders than the SD method. On the other hand, the false-positive probability of HD method is much higher than that of SD method. It is because several bits are flipped in the HD method by white Gaussian noise. Since the SD method calculates Z_{SD} according to the distribution of S'_j, the false-positive probability is not so degraded. However, such a threshold Z_{SD} is not always valid because the Gaussian assumption of the distribution of S'_j becomes invalid when SNR is decreased. Under the constant number of colluders, the number of detected colluders and the false-positive probability are evaluated by changing the amount of noise. Figures 3 and 4 show the results when the number of colluders is 10. Regardless of the amount of noise, the traceability of HD method is better than that of SD method. It is noticed that the false-positive probability approaches 10^{-5} for both methods when SNR is increased. It is because the probability is

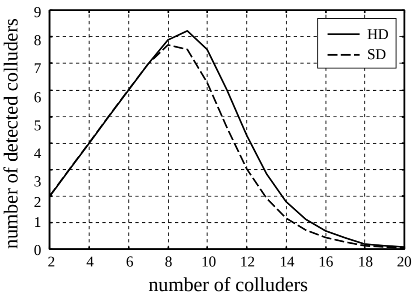


Fig. 1. The number of detected colluders when SNR is 8 [dB].

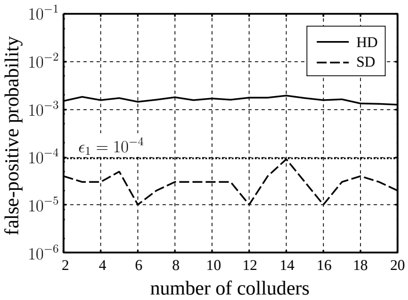


Fig. 2. The false-positive probability when SNR is 8 [dB].

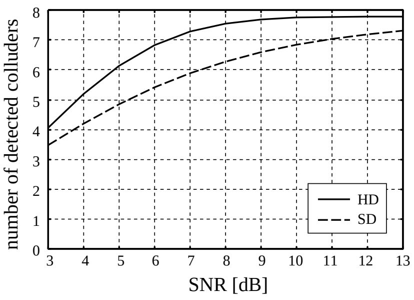


Fig. 3. The number of detected colluders versus SNR when $\tilde{c} = 10$.

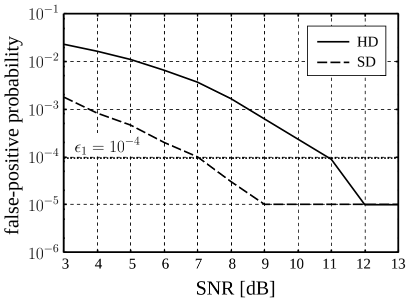


Fig. 4. The false-positive probability versus SNR when $\tilde{c} = 10$.

10^{-5} when a pirated copy is not distorted by noise under the above conditions. From these results, the use of CLT seems to be valid only when the amount of noise is very small. Meanwhile, the result in Fig.4 shows the significant property of both methods such that the probability of false-positive is increased with the decrease of SNR. It also means that the increase of the probability is strongly dependent on the number of flipped bits at a pirated codeword. It is remarkable that the increasing rate of the probability of false-positive for the conventional tracing algorithms that use a threshold to determine the guilty must be similar to the above results because such a threshold is independent on the noise and bit flips.

4 Proposed Tracing Algorithm

The number of flipped bits is increased with the noise energy because the probabilities $\Pr(\hat{y}_i = 1 \cap e_i < -1)$ and $\Pr(\hat{y}_i = -1 \cap e_i > 1)$ become non-negligible. In such a case, the designed threshold Z_{HD} is not valid. In the HD method, the degraded signal $\hat{y}'_i$ is classified into only two symbols "-1" and "1" if $\hat{y}'_i$ is more than 0 or not. Considering the variance σ_e^2 of Gaussian noise, the classification should be adaptively modified by a threshold $T_{\sigma_e^2}$ that classifies $\hat{y}'_i$ into three symbols "-1", "1", and "0".

4.1 Channel Estimation

After extracting the fingerprint signal from a pirated copy, we estimate the variance of Gaussian noise using the extracted analog values. Using the variance σ_{SD}^2, the threshold $T_{\sigma_e^2}$ is obtained accordingly.

If $|y'_i| \geq 1$, then the absolute value of the amplitude of noise is estimated as

$$|e_i| = |\hat{y}'_i| - 1 \ , \tag{9}$$

because

$$\Pr(\hat{y}_i = 1 | \hat{y}'_i > 1) \gg \Pr(\hat{y}_i = -1 | \hat{y}'_i > 1) \ , \tag{10}$$

and

$$\Pr(\hat{y}_i = -1 | \hat{y}'_i < -1) \gg \Pr(\hat{y}_i = 1 | \hat{y}'_i < -1) \ . \tag{11}$$

The probability P_{err} that the estimation of Eq.(9) is failed can be calculated from $\Pr(\hat{y}_i = -1|\hat{y}'_i > 1)$ and $\Pr(\hat{y}_i = 1|\hat{y}'_i < -1)$, and is represented by

$$P_{err} = \frac{1}{2} \Pr\big((e_i < -2) \cup (e_i > 2)\big) = \frac{1}{2} \Pr(|e_i| > 2) \ . \tag{12}$$

Considering the property of AWGN channel, the probability $\Pr(|e_i| > 2)$ can be calculated by

$$\Pr(|e_i| > 2) = \mathrm{erfc}\left(\frac{2}{\sqrt{2\sigma_e^2}}\right) \ , \tag{13}$$

where σ_e^2 is the variance of noise. Hence, when σ_e^2 is not very large, P_{err} is negligible and the estimation of Eq.(9) is valid. In such a case, σ_e^2 can be calculated by

$$\sigma_e^2 = \frac{\sum_{i \in \{|\hat{y}'_i| \geq 1\}} (e_i - \overline{e})^2}{L_e} \ , \tag{14}$$

where $\overline{e}$ is the average value of $e_i, i \in \{|\hat{y}'_i| \geq 1\}$ and L_e is the number of $\hat{y}'_i$ satisfies $|\hat{y}'_i| \geq 1$.

When the variance σ_e^2 of Gaussian noise is very small, the probability $\Pr(\hat{y}'_i > 0|\hat{y}_i = -1)$ is negligible, and then the threshold for the classification is $T_{\sigma_e^2} = 0$. Suppose that the probability is non-negligible. For a given threshold $T_{\sigma_e^2}$, the probability P_{flip} that at least one symbol is flipped is calculated by

$$P_{\mathit{flip}} = \frac{1}{2} \mathrm{erfc}\left(\frac{T_{\sigma_e^2}}{\sqrt{2\sigma_e^2}}\right) \ . \tag{15}$$

If the code length is L, the average number of flipped bits is $P_{\mathit{flip}}L$. By transforming Eq.(15), the threshold $T_{\sigma_e^2}$ is calculated as follows.

$$T_{\sigma_e^2} = \sqrt{2\sigma_e^2} \cdot \mathrm{erfc}^{-1}\big(2P_{\mathit{flip}}\big) \tag{16}$$

Using Eq.(16), we can calculate the threshold $T_{\sigma_e^2}$ for the given probability P_{flip}.

4.2 Adaptive Tracing Algorithm

It is reasonable to apply the HD method for $|\hat{y}'_i| \geq T_{\sigma_e^2}$ because they are judged as the symbols "-1" or "1" with high probability $1 - P_{flip}$. Hence, such elements are reliable to calculate correlation scores, while the other elements, $|\hat{y}'_i| < T_{\sigma_e^2}$, are unreliable. Considering the property derived from the result in Sect.3.3, the unreliable elements should be avoided in order to exclude the bit flips in a pirated codeword. Thus, we replace the elements of pirated codeword $\hat{y}'_i$ with $Y_i^{(1)}$ as follows.

$$Y_i^{(1)} = \begin{cases} 1 & (\hat{y}'_i \geq T_{\sigma_e^2}) \\ -1 & (\hat{y}'_i \leq -T_{\sigma_e^2}) \\ 0 & (|\hat{y}'_i| < T_{\sigma_e^2}) \end{cases} \tag{17}$$

Notice that the above replacement implies the binary erasure channel (BEC).

Let $L^{(1)}$ be the number of $|\hat{y}'_i| \geq T_{\sigma_e^2}$ and $Z_{HD}^{(1)}$ be the proper threshold for the determination of guilty. We first calculate the correlation score $S_j^{(1)}$:

$$S_j^{(1)} = \sum_{i=1}^{L} Y_i^{(1)} U_{j,i} \ . \tag{18}$$

Then, $L^{(1)}$ is derived by counting the number of elements $|\hat{y}'_i| \geq T_{\sigma_e^2}$, which is corresponding to the variance of $S_j^{(1)}$. For the given probability $\epsilon_2^{(1)}$ such that an j-th innocent user gets accused, the threshold $Z_{HD}^{(1)}$ is calculated as follows.

$$Z_{HD}^{(1)} = \sqrt{2L^{(1)}} \cdot \mathrm{erfc}^{-1}\big(2\epsilon_2^{(1)}\big) \tag{19}$$

Because of the randomness of the Gaussian noise, the reliable elements is regarded as the elements of sub-codeword with length $L^{(1)}$. The traceability of the above method is lower than the original HD method because the length of sub-codeword is reduced to $L^{(1)}(\leq L)$, though the increase of the false-positive probability is limited. We denote this method by "method I".

At the method I, only reliable elements $|\hat{y}'_i| \geq T_{\sigma_e^2}$ are selected for the tracing algorithm. It does not mean that the other elements are useless for the judgment. They also contain useful information to improve the traceability though they may increase the probability of false-positive. In order to extract as much information as possible without sacrificing the probability of false-positive, we propose a new tracing algorithm which consists of two stages. First, suspicious users are listed up using the method I by setting $\epsilon_2^{(1)}$ higher to allow the false-positive at this stage. Then, the elements $|\hat{y}'_i| < T_{\sigma_e^2}$ are classified into two symbols "-1" and "1", and the others are changed 0. The replaced elements $Y_i^{(2)}$ are represented as follows;

$$Y_i^{(2)} = \begin{cases} 1 & (0 \leq \hat{y}'_i < T_{\sigma_e^2}) \\ -1 & (-T_{\sigma_e^2} < \hat{y}'_i < 0) \\ 0 & (|\hat{y}'| \geq T_{\sigma_e^2}) \end{cases} \tag{20}$$

Only for the suspicious users whose scores are $S_j^{(1)} > Z_{HD}^{(1)}$, the correlation scores $S_j^{(2)}$ are calculated as follows.

$$S_j^{(2)} = S_j^{(1)} + \sum_{i=1}^{L} Y_i^{(2)} U_{j,i} \tag{21}$$

Finally, j-th user is judged guilty if $S_j^{(2)} > Z_{HD}$, where the threshold Z_{HD} is given by Eq.(7). We denote this method by "method Ⅱ".

In the method Ⅱ, we first detect suspicious users from all users using the sub-codeword under the criterion that their sub-codewords retain high correlation with that of pirated codeword. The sub-codeword of pirated codeword is composed of reliable elements $|\hat{y}'_i| \geq T_{\sigma_e^2}$ and the number of bit flips caused by the additive noise is only $P_{flip}L$ in the sub-codeword. If $P_{flip}L$ is small, the number of innocent users involved in the detected suspicious users is expected to be $\epsilon_2^{(1)}N$. In such a case, even if some innocent users are accidentally detected at the first stage, the second stage excludes such innocent users with high probability. In addition, without loss of generality, the following relation is satisfied.

$$\Pr\left(S_j^{(2)} > Z_{HD}|S_j^{(1)} > Z_{HD}^{(1)}, j \in \mathcal{I}\right) > \Pr(S_j > Z_{HD}, j \in \mathcal{I}) \ , \qquad (22)$$

where $\mathcal{I}$ stands for a set of innocent users. Therefore, the method II can reduce the probability of false-positive effectively.

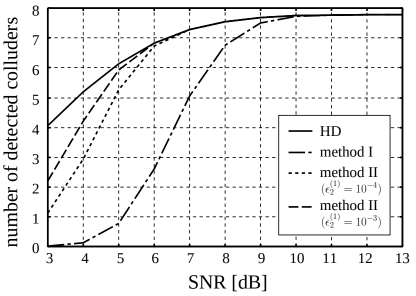
The performance of method Ⅱ depends on the selection of $\epsilon_2^{(1)}$ and P_{flip} as the number of suspicious users detected by the tracing algorithm is controlled by these parameters. Remember that it is desirable to keep the number of bit flips $P_{flip}L$ in a sub-codeword as small as possible from the result in Sect.3.3.

5 Experimental Results

We implement the proposed tracing algorithms and evaluate the collusion-resistance. Under the marking assumption, $\tilde{c}$ codewords are randomly chosen and majority attack is performed to produce a pirated codeword $\hat{\boldsymbol{y}}$. After the collusion attack, white Gaussian noise is added to the pirated codeword $\hat{y}_i'$ and try to detect as many colluders as possible from the degraded codeword $\hat{y}_i'$. The length of codeword is $L = 10^4$ and the range of bias distribution p_i is given by setting $t = 0.000167$ $(c = 20)$. The number of users is $N = 10^4$, and the false-positive probability is $\epsilon_1 = 10^{-4}$. For the design of the threshold $T_{\sigma_e^2}$, the probability is fixed to $P_{flip} = 10^{-4}$, which average number of flipped bits is $P_{flip}L = 1$. The number of trials for Monte-Carlo simulation is 10^5.

First, the number of detected colluders is evaluated for various SNR of AWGN channel with a fixed number of colluders $\tilde{c} = 10$. As evaluated in Sect.3.3, the number of detected colluders of SD method is lower than that of HD method, so we compare the performance of proposed methods with the HD method, which results are shown in Fig.5 and Fig.6. We can see that the performance of the method I is much lower than the others. It is because of the short code length $L^{(1)} \leq L$. The method Ⅱ improves the performance compared with the method I. When the threshold $Z_{HD}^{(1)}$, which value is dependent on the given probability $\epsilon_2^{(1)}$, is small, the number of suspicious users is increased, and hence, the number of detectable colluders is improved as shown by the two cases $\epsilon_2^{(1)} = 10^{-3}$ and $\epsilon_2^{(1)} = 10^{-4}$.

For the comparison with the HD method, the probabilities of false-positive are shown in Fig.6. Although the probabilities of method I and method Ⅱ are slightly growing up with the increase of the amount of noise, the increasing rate is much smaller than that of HD method. Compared with method Ⅱ, we can see that the false-positive probability of method I is monotonically growing up. It is because of the following reason. P_{err} becomes large with the decrease of



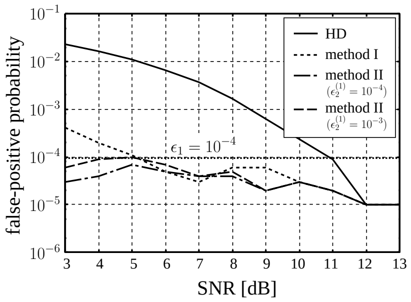
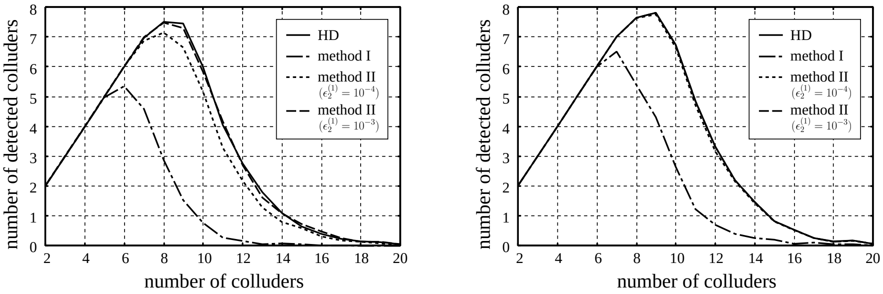


Fig. 5. The number of detected colluders, where $\tilde{c} = 10$ and $\epsilon_1 = 10^{-4}$.

Fig. 6. The probability of false-positive, where $\tilde{c} = 10$ and $\epsilon_1 = 10^{-4}$.



(a) SNR 5 [dB]

(b) SNR 6 [dB]

Fig. 7. Comparison of the number of detected colluders for various SNR.

SNR, and hence, the variance σ_e^2 estimated by Eq.(14) becomes smaller than the actual one. It causes the error on the estimation of the threshold $T_{\sigma_e^2}$. Since the derived threshold is smaller than the actual one, the probability P_{flip} becomes large. As the result, the number of flipped bits are increased and accordingly the probability of false-positive is increased. It is remarkable that the error on the estimation of σ_e^2 is almost canceled at the final determination of guilty in method Ⅱ, which is conformed by the experimental results.

As the reference data, the comparison of the number of detected colluders is shown in Fig.7 by changing SNR, where the number of trials is 10^2 times. When SNR is 5 [dB], the performance of method Ⅱ with $\epsilon_2^{(1)} = 10^{-3}$ is better than that of method Ⅱ with $\epsilon_2^{(1)} = 10^{-4}$ because of the difference in the number of detected suspicious users. When SNR is 6 [dB], no remarkable difference of the performance is appeared, and they are approaching to the lines of HD method. Referring to the results in Fig.6, it is confirmed that the false-positive probability of method Ⅱ is strongly dependent on the design of $\epsilon_2^{(1)}$.

For the evaluation of stability of false-positive probability, the number of colluders $\tilde{c}$ is increased to produce a pirated codeword, and after the addition of noise, it is input to the proposed tracing algorithm. The probability of false-

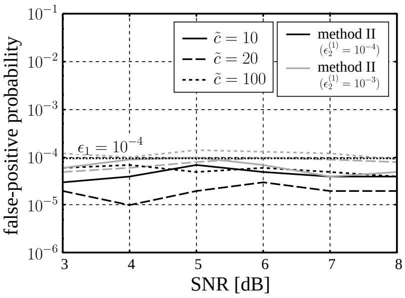


Fig. 8. The probability of false-positive for various number of colluders.

positive for various number of colluders is plotted in Fig.8. We can see that the probabilities are almost within a small range even if the number of colluders is changed. We also evaluate the probability of false-positive for various kinds of collusion attacks, which results are shown in Table 1. The table confirms that the probability of false-positive is not dependent on the attack strategy.

The performance of the proposed tracing algorithm is further evaluated for various kinds of parameters. Table 2 and Table 3 show the number of detected colluders and the probability of false-positive under a constant number of colluders $\tilde{c}$ when the allowable numbers of users in a fingerprinting system are $N = 10^5$ and $N = 10^6$, respectively. Due to the limitation of computational resources, the number of trials for Monte Carlo simulation is 10^5 and 10^4 for $N = 10^5$ and 10^6, respectively. In addition, we use the probabilities $\epsilon_1 = 10^{-4}$ and $\epsilon_1 = 10^{-3}$ to keep the precision of the derived probability of false-positive. We set $\epsilon_2^{(1)}$ under the policy that the number of innocent users in the detected suspicious users is 10 in average. From these tables, we can see that the proposed tracing algorithm with the above given parameters detects many colluders with less innocent users, and the probability of false-positive is very close to the designed probability ϵ_1. The comparisons of the number of detected colluders and the probability of false-

Table 1. Comparison of false-positive probability for various kinds of collusion attacks, where $N = 10^4$, $\tilde{c} = 10$, $L = 10000$, $\epsilon_2^{(1)} = 10^{-3}$, and $\epsilon_1 = 10^{-4}$.

SNR	tracing	collusion attack				
[dB]	algorithm	majority	minority	random	All-0	All-1
5	HD	111.6×10^{-4}	111.9×10^{-4}	105.2×10^{-4}	113.3×10^{-4}	110.6×10^{-4}
	method Ⅱ	0.9×10^{-4}	1.2×10^{-4}	0.7×10^{-4}	1.2×10^{-4}	0.7×10^{-4}
8	HD	17.1×10^{-4}	16.0×10^{-4}	17.2×10^{-4}	16.6×10^{-4}	15.6×10^{-4}
	method Ⅱ	0.5×10^{-4}	1.0×10^{-4}	0.7×10^{-4}	0.8×10^{-4}	0.4×10^{-4}
13	HD	0.1×10^{-4}	0.7×10^{-4}	0.2×10^{-4}	0.2×10^{-4}	0.4×10^{-4}
	method Ⅱ	0.1×10^{-4}	0.7×10^{-4}	0.2×10^{-4}	0.2×10^{-4}	0.4×10^{-4}

Table 2. The number of detected colluders under a constant number of colluders $\tilde{c} = 10$ when the allowable number of users is expanded, where the code length is $L = 10^4$.

(a) $N = 10^5$, $\epsilon_2^{(1)} = 10^{-4}$, and $\epsilon_1 = 10^{-4}$

SNR [dB]	HD	method Ⅱ
5	4.62	4.17
8	6.21	6.21
13	6.50	6.50

(b) $N = 10^6$, $\epsilon_2^{(1)} = 10^{-5}$, and $\epsilon_1 = 10^{-3}$

SNR [dB]	HD	method Ⅱ
5	4.57	4.12
8	6.16	6.16
13	6.45	6.45

Table 3. The probability of false-positive under a constant number of colluders $\tilde{c} = 10$ when the allowable number of users is expanded, where the code length is $L = 10^4$.

(a) $N = 10^5$, $\epsilon_2^{(1)} = 10^{-4}$, and $\epsilon_1 = 10^{-4}$

SNR [dB]	HD	method Ⅱ
5	871.6×10^{-4}	3.5×10^{-4}
8	135.7×10^{-4}	3.7×10^{-4}
13	1.1×10^{-4}	1.1×10^{-4}

(b) $N = 10^6$, $\epsilon_2^{(1)} = 10^{-5}$, and $\epsilon_1 = 10^{-3}$

SNR [dB]	HD	method Ⅱ
5	910.8×10^{-3}	2.6×10^{-3}
8	146.5×10^{-3}	2.2×10^{-3}
13	0.5×10^{-3}	0.5×10^{-3}

positive are shown in Fig.9 and Fig.10, respectively. The solid line represents the result of the case that the length is $L = 10000$ and the number of colluders is $\tilde{c} = 10$, the dashed line is the case with $L = 5000$ and $\tilde{c} = 7$, and the dotted line is the case with $L = 2000$ and $\tilde{c} = 4$. From Fig.9, it is confirmed that the performance of method II is degraded from that of HD method when SNR drops to less than 6 [dB]. The probability of false-positive is, however, much smaller than that of HD method, and it almost keeps within a small range even if the length is changed.

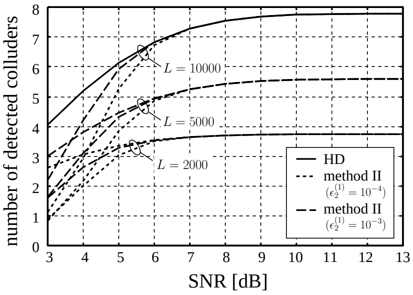


Fig. 9. Comparison of the number of detected colluders, where $N = 10^4$ and $\tilde{c} = 10, 7, 4$ for the codes with length $L = 10000, 5000, 2000$, respectively.

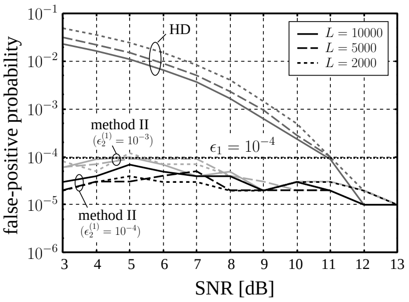


Fig. 10. Comparison of the false-positive probability, where $N = 10^4$ and $\tilde{c} = 10, 7, 4$ for the codes with length $L = 10000, 5000, 2000$, respectively.

6 Conclusion

In this paper, we relaxed the marking assumption to consider more realistic situation. In our attack model, a pirated codeword is modulated by BPSK, and is degraded by additive white Gaussian noise after performing collusion attack. Considering the watermarking technique, the extracted codeword from the pirated copy is represented by analog values. To accommodate with the degradation caused by the noise, the proposed tracing algorithm first estimates the amount of noise injected to a channel, and then, detects as many colluders as possible. In order not to increase the probability of false-positive, the proposed algorithm classify the elements of the codeword into reliable ones and the others and detect suspicious users using the former ones with the threshold calculated under the Gaussian assumption of the correlation score. Then, among the suspicious users, the proposed algorithm narrow down the suspicious users using the whole codeword with the corresponding threshold. From the simulation results, it is confirmed that the proposed tracing algorithm can detect many colluders with less innocent users.

Acknowledgment

This research was partially supported by the Ministry of Education, Culture, Sports Science and Technology, Grant-in-Aid for Young Scientists (B) (21760291), 2010.

References

1. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory 44(5), 1897–1905 (1998)
2. Škorić, B., Vladimirova, T.U., Celik, M., Talstra, J.C.: Tardos fingerprinting is better than we thought. IEEE Trans. Inform. Theory 54(8), 3663–3676 (2008)
3. Furon, T., Guyader, A., Cérou, F.: On the design and optimization of Tardos probabilistic fingerprinting codes. In: IH 2008. LNCS, vol. 5284, pp. 341–356. Springer, Heidelberg (2008)
4. Guth, H.J., Pfitzmann, B.: Error- and collusion-secure fingerprinting for digital data. In: IH 1999. LNCS, vol. 1768, pp. 134–145. Springer, Heidelberg (2000)
5. Kuribayashi, M., Morii, M.: Systematic generation of Tardos's fingerprinting codes. IEICE Trans. Fundamentals E93-A(2), 508–515 (2009)
6. Nuida, K.: Making collusion-secure codes (more) robust against bit erasure. In: eprint. 2009-549 (2009)
7. Nuida, K., Fujitu, S., Hagiwara, M., Kitagawa, T., Watanabe, H., Ogawa, K., Imai, H.: An improvement of discrete Tardos fingerprinting codes. Designs, Codes and Cryptography 52(3), 339–362 (2009)
8. Safavi-Naini, R., Wang, Y.: Collusion secure q-ary fingerprinting for perceptual content. In: DRM 2001. LNCS, vol. 2320, pp. 57–75. Springer, Heidelberg (2002)
9. Tardos, G.: Optimal probabilistic fingerprint codes. J. ACM 55(2), 1–24 (2008)
10. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion resistant fingerprinting for multimedia. IEEE Signal Processing Mag. pp. 15–27 (2004)