A New Soft Decision Tracing Algorithm for Binary Fingerprinting Codes

Minoru Kuribayashi

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

Abstract. The performance of fingerprinting codes has been studied under the well-known marking assumption. In a realistic environment, however, a pirated copy will be distorted by an additional attack. Under the assumption that the distortion is modeled as AWGN, a soft decision method for a tracing algorithm has been proposed and the traceability has been experimentally evaluated. However, the previous soft decision method works directly with a received signal without considering the communication theory. In this study, we calculate the likelihood of received signal considering a posterior probability, and propose a soft decision tracing algorithm considering the characteristic of Gaussian channel. For the estimation of channel, we employ the expectation-maximization algorithm by giving constraints under the possible collusion strategies. We also propose an equalizer to give a proper weighting parameter for calculating a correlation score.

1 Introduction

Digital fingerprinting [14] is used to trace illegal users, where a unique ID known as a digital fingerprint is embedded into a content before distribution. When a suspicious copy is found, the owner can identify illegal users by extracting the fingerprint. Since each user purchases a content involving his own fingerprint, the fingerprinted copy slightly differs with each other. Therefore, a coalition of users will combine their differently marked copies of the same content for the purpose of removing/changing the original fingerprint. To counter this threat, coding theory has produced a number of collusion resistant codes under the well-known principle referred to as the marking assumption.

Tardos [13] has proposed a probabilistic fingerprinting code which has a length of theoretically minimal order with respect to the number of colluders. Theoretical analysis about the Tardos code yields more efficient probabilistic fingerprinting codes improving the traceability, code length, and so on. Among the variants of the Tardos code, Nuida et al. [10] studied the parameters to generate the codewords of the Tardos code which are expressed by continuous distribution, and presented a discrete version in an attempt to reduce the code length and the required memory amount without degrading the traceability.

T. Iwata and M. Nishigaki (Eds.): IWSEC 2011, LNCS 7038, pp. 1–15, 2011.
© Springer-Verlag Berlin Heidelberg 2011

It is reported in [2] that a correlation sum calculated in a tracing algorithm is expected to be Gaussian distribution based on the Central Limit Theorem (CLT). Using the Gaussian approximation, the code length is further shortened under a given false-positive probability. The results are supported and further analyzed by Furon et al. [3], and the validity is experimentally evaluated in [8]. In [12], it is shown that the tails of the distribution follow a power law which depends on the collusion strategy. Independent of the strategy, the right tail falls off faster than the left tail.

Recently, the relaxation of the marking assumption has been employed in the analysis of the Tardos code and its variants [5],[6],[7],[9]. In [7], a pirated copy is produced by collusion attack and it is further distorted by additive white Gaussian noise (AWGN). Considering the distortion, two kinds of tracing algorithms are proposed; one rounds each element of codeword into binary digit before calculating a correlation score, and the other directly calculates the score from the distorted codeword. The former is called a hard decision method, and the latter, a soft decision method. In [6], it is reported that the probability of false-positive for the Tardos code is considerably increased in the amount of noise while that for the Nuida code is not sensitive against the noise. However, the soft decision method does not utilize the analog signals to maximize the performance of a detector. It merely calculates a correlation score directly from the received signal without the consideration of a posterior probability.

In this paper, we propose a soft decision tracing algorithm considering a posterior probability of codeword extracted from a pirated copy. We assume that a codeword is produced by a certain collusion strategy based on the marking assumption and is distorted by additive white Gaussian noise. Depending on the collusion strategy, the probability that an i-th bit becomes 1 is slightly/greatly changed from the original probability, namely 0.5. In order to estimate the probability as well as the variance of the Gaussian noise, the Expectation-Maximzation(EM) algorithm is used in this paper. Generally, the EM algorithm is not assured to find a global optimum whose estimated values are well-matched with actual ones. By giving some constraints on the parameters estimated by the EM algorithm, we improve the accuracy to find the global optimum. Using the estimated parameters, we calculate a new correlation sum based on the posterior probability. If the sum exceeds a specific threshold, the corresponding candidate is judged guilty. Based on the CLT, the variance of the sum is derived from a Monte Carlo simulator and the threshold for judgment is calculated by a given false-positive probability. The validity of the threshold is also evaluated by the rare event simulation method proposed in [5]. We further study the bias in the calculation of the correlation score, and propose an equalizer to cancel the bias by giving a weight on each score.

The experimental results reveal the following properties. 1: When the EM algorithm fails to estimate the conditions of Gaussian channel, the performance of the proposed method without the equalizer is degraded with the increase of SNR. 2: The proposed method with the equalizer outperforms the method without it. Especially for the cryptographic collusion strategy [3], we get a drastic improvement from the conventional methods. 3. The total false-positive probability is almost stable against the changes of SNR, and is slightly affected by a collusion strategy if the threshold is designed under the Gaussian assumption.

2 Preliminaries

In this section, probabilistic fingerprinting codes are reviewed, and the related works are briefly introduced.

2.1 Probabilistic Fingerprinting Code

Tardos [13] has proposed a probabilistic c-secure code which has a length of theoretically minimal order with respect to the number of colluders. The binary codewords of length L are arranged as an $N \times L$ matrix $\boldsymbol{X}$, where N is the number of users and each element $X_{j,i} \in \{0,1\}$ in the matrix is the i-th element of j-th user's codeword. The element $X_{j,i}$ is generated from an independently and identically distributed random number with a probability p_i such that $\Pr[X_{j,i} = 1] = p_i$ and $\Pr[X_{j,i} = 0] = 1 - p_i$. This probability p_i referred to as the *bias distribution* follows a certain continuous distribution represented by $f(p)$:

$$f(p) = \frac{1}{\pi\sqrt{p(1-p)}}. \tag{1}$$

Assuming that the number of colluders is at most c, the minimum length L for a constant and tiny error probability is theoretically derived. The maximum allowed probability of accusing a fixed innocent user is denoted by ϵ_1, and the total false positive probability by $\eta = 1 - (1-\epsilon_1)^{N-c} \approx N\epsilon_1$. The false negative probability denoted by ϵ_2 is coupled to ϵ_1 according to $\epsilon_2 = \epsilon_1^{c/4}$.

Nuida et al. [10] proposed a specific discrete distribution introduced by a discrete variant [11] of Tardos code that can be tuned for a given number c of colluders. The bias distribution is called "Gauss-Legendre distribution" due to the deep relation to Gauss-Legendre quadrature in numerical approximation theory (see [10] for detail). Except for the bias distribution, the Nuida code employs the same encoding mechanism as the Tardos code.

Let L be a code length of a fingerprinting code. Suppose that $\tilde{c}(\leq c)$ malicious users out of N users are colluded, and they produce a pirated codeword $\boldsymbol{y} = (y_1, \ldots, y_L)$, $y_i \in \{0,1\}$. A tracing algorithm first calculates a score $S_i^{(j)}$ for i-th bit of j-th user using a real-valued function $U_{j,i}$, and then sums them up as the total score $S^{(j)} = \sum_{i=0}^{L} S_i^{(j)}$ of j-th user.

$$S^{(j)} = \sum_{i=1}^{L} S_i^{(j)} = \sum_{i=1}^{L} y_i U_{j,i}, \tag{2}$$

where

$$U_{j,i} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}} & (X_{j,i} = 1) \\ -\sqrt{\frac{p_i}{1-p_i}} & (X_{j,i} = 0). \end{cases} \tag{3}$$

Because the above correlation sum adds the score $S_i^{(j)}$ only when $y_i = 1$, half of the elements in a pirated codeword is discarded. Considering the symmetry, Škorić et al. [2] proposed a symmetric version of the correlation score by substituting $\hat{y}_i = 2y_i - 1 \in \{-1, 1\}$ for y_i in Eq.(2).

For the Tardos code, if the sum $S^{(j)}$ exceeds a threshold Z, the j-th user is determined as guilty. Such a tracing algorithm is called "catch-many" type explained in [14]. By decoupling ϵ_1 from ϵ_2, the tracing algorithm can detect more colluders under a constant ϵ_1 and L. For the Nuida code [10], its original tracing algorithm outputs only one guilty user whose score becomes maximum, which type is called "catch-one". Due to the similarity with the Tardos code, the catch-many tracing algorithm of the Tardos code can be applied to the Nuida code. The report in [6] stated that the performance of the Nuida code is better than that of the Tardos code when the catch-many tracing algorithm is used. Under a same code length and a same number of colluders, it is experimentally measured that the correlation sum of the Nuida code is higher than that of the Tardos code. It is remarkable that the false-positive probability of the Nuida code is stable 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 if a threshold is calculated under the Gaussian approximation for the correlation score. In this paper, the validity of the previous tracing algorithms is discussed from the Nuida code point of view, which does not limit the use of proposed method for the Tardos code.

2.2 Attack Model

Under the marking assumption, colluders can select an arbitrary bit for such elements that a bit embedded into the segments of their copies is different. Based on an attack strategy, various collusion strategies under the marking assumption could be selected by colluders. Among them, there are 5 major types:

- majority(maj): If the sum of i-th bit exceeds $\tilde{c}/2$, $y_i = 1$; otherwise, $y_i = 0$.
- minority(min): If the sum of i-th bit exceeds $\tilde{c}/2$, $y_i = 0$; otherwise, $y_i = 1$.
- random(ran): $y_i \in_R \{0, 1\}$
- all-0: $y_i = 0$
- all-1: $y_i = 1$

In [5], the collusion attack is described by the parameter vector: $\boldsymbol{\theta} = (\theta_0, \cdots, \theta_{\tilde{c}})$ with $\theta_\rho = \Pr_y[1|\Phi = \rho]$, where the random variable $\Phi \in \{0, \cdots, \tilde{c}\}$ denotes the number of symbol "1" in the colluders' copies at a given index. Furthermore, the Worst Case Attack(WCA) is defined as the collusion attack minimizing the rate of the code, or equivalently, the asymptotic positive error exponent. For example, when $\tilde{c} = 5$, the parameter vector of WCA is given by $\theta^\star = (0, 0.594, 0.000, 1.00, 0.406, 1)$.

On the other hand, the attack strategies are not limited to the above types in a realistic situation such that a codeword is binary and each bit is embedded into one of segments of a digital content without overlapping using a robust watermarking scheme. It is reasonable to assume that each bit is embedded into a segment using an antipodal signal: $\hat{X}_{j,i} = 2X_{j,i} - 1$, namely it is binary phase shift keying(BPSK) modulation. In this case, colluders can apply the other attack strategy at the detectable positions. Since each bit of codeword of $\hat{\boldsymbol{y}}$ is one of $\{-1, 1\}$ after the BPSK modulation, it is possible for colluders to alter the signal amplitude of each element from the signal processing point of view. One simple example is averaging attack that $\hat{y}_i = \sum \hat{X}_{j,i}/c$, we call this attack "average(ave)". Considering the removal of fingerprint signal, a worst case may be $\hat{y}_i = 0$. At the detectable position, it is sufficient to average only two segments whose $\hat{X}_{j,i}$ are different with each other, which attack is denoted by "average2(ave2)".

Even if a robust watermarking method is used to embed the binary fingerprinting code into digital contents, it must be degraded by attacks. For convenience, the distortion is modeled as AWGN in this study. So, we assume that a pirated copy is produced by one of the above collusion strategies and is further distorted by the Gaussian noise.

2.3 Conventional Tracing Algorithm

Assuming that the pirated codeword $\hat{\boldsymbol{y}}$ is transmitted over AWGN channel. Then, the codeword extracted from a pirated copy is represented by analog value:

$$\boldsymbol{y'} = \hat{\boldsymbol{y}} + \boldsymbol{e} = (\hat{y}_1 + e_1, \dots, \hat{y}_L + e_L), \tag{4}$$

because of the addition of noise $\boldsymbol{e}$ that follows $N(0, \sigma_e^2)$. If a tracing algorithm strictly follows the definition, each extracted symbol of the pirated codeword should be rounded into a bit $\{-1, 1\}$ when the symmetric version of the tracing algorithm is used. Because of the rounding operation, this procedure is called a hard decision (HD) method in [7] and [6]. On the other hand, it is possible to directly calculate the correlation sum $S^{(j)}$ from the distorted pirated codeword $\boldsymbol{y'}$, which procedure is called a soft decision (SD) method. A soft decoding method is very beneficial in error correcting code, so it is worthy to try for fingerprinting. However, in the SD method, the likelihood of the received signal is not considered to maximize the traceability. It is strongly required for the soft decision method to calculate the correlation score based on the information theoretic analysis.

3 Proposed Tracing Algorithm

The proposed tracing algorithm first estimates the amount of noise involved in a pirated copy and then measures the likelihood of each symbol of pirated copy. Using the likelihood, the correlation score is calculated and guilty users are identified with a constant false probability ϵ_1.

3.1 Channel Estimation

The accurate estimation of the Gaussian channel can maximize the performance of tracing algorithm. The estimator proposed in [7] does not make use of all the available samples, but only half samples in average. In addition, it only estimates the variance σ_e^2 of Gaussian noise. In this paper, we estimate the probability distribution function that is regarded as a Gaussian mixture model.

If a collusion strategy is based on the marking assumption, each symbol of a pirated codeword is $\hat{y}_i \in \{-1, 1\}$. Here, the probability $\Pr[\hat{y}_i = 1]$ is not always equal to $\Pr[\hat{y}_i = -1]$. So, the probability distribution function $\mathit{pdf}(y'_i)$ is represented by

$$\mathit{pdf}(y'_i) = aN(y'_i; 1, \sigma_e^2) + (1-a)N(y'_i; -1, \sigma_e^2), \tag{5}$$

where $a \geq 0$ and

$$N(y'_i; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(y'_i - \mu)^2}{2\sigma^2} \right). \tag{6}$$

Under the relaxed version of the marking assumption, the value of $\hat{y}_i$ is not limited to these two symbols. Hence, the probability distribution function can be a mixture of several Gaussian components, and in general, it is denoted by

$$\mathit{pdf}(y'_i) = \sum_{k=1}^{m} a_k N(y'_i; \mu_k, \sigma_k^2), \tag{7}$$

where m is the number of Gaussian components, and $\sum_{k=1}^{m} a_k = 1$ and $a_k \geq 0$.

Thanks to the EM algorithm [1], we can derive unknown parameters a_k, μ_k, and σ_k^2 from $\boldsymbol{y'}$ and $\mathit{pdf}(y'_i)$. The EM algorithm is a well-established maximum likelihood algorithm for fitting a mixture model to a set of training data. The algorithm is an iterative method which alternates between performing an expectation(E)-step and a maximization(M)-step. The E-step computes the expectation of the log-likelihood evaluated from the current estimate for the latent variables, and the M-step computes parameters maximizing the expected log-likelihood found on the E-step. Because it is very popular to estimate the parameters of Gaussian mixture model using the EM algorithm, we only describe the procedure to estimate the unknown parameters in this paper (see [1] for detail).

Let $\boldsymbol{\Theta}$ be a vector of unknown parameters a_k, μ_k, and σ_k^2. The log-likelihood function $L(\boldsymbol{y'}, \boldsymbol{\Theta})$ with respect to $\boldsymbol{y'}$ is represented by

$$L(\boldsymbol{y'}, \boldsymbol{\Theta}) = \log \Pr[\boldsymbol{y'}, \boldsymbol{\Theta}] = \sum_{i=1}^{L} \log \left(\sum_{k=1}^{m} a_k N(y'_i; \mu_k, \sigma_k^2) \right). \tag{8}$$

The goal is to maximize the posterior probability of the parameters $\boldsymbol{\Theta}$ from $\boldsymbol{y'}$ in the presense of hidden parameters $\boldsymbol{\xi}$. The EM algorithm seeks to find the maximum likelihood estimate of $L(\boldsymbol{y'}, \boldsymbol{\Theta})$ by iteratively applying the following two steps:

- E-step: Calculate the conditional distribution of $\xi_{k,i}$ under the current estimate of the parameters $\boldsymbol{\Theta}^{(t)}$:

$$\xi_{k,i} = \frac{a_k N(y'_i; \mu_k, \sigma_k^2)}{\sum_{h=1}^{m} a_h N(y'_i; \mu_h, \sigma_h^2)} \tag{9}$$

- M-step: Calculate the estimated parameters $\boldsymbol{\Theta}^{(t+1)}$ that maximize the expected value of $L(\boldsymbol{y'}, \boldsymbol{\Theta}^{(t+1)})$ using $\boldsymbol{\xi}$:

$$a_k = \frac{1}{N} \sum_{i=1}^{L} \xi_{k,i}, \tag{10}$$

$$\mu_k = \frac{\sum_{i=1}^{L} \xi_{k,i} y'_i}{\sum_{i=1}^{L} \xi_{k,i}}, \tag{11}$$

and

$$\sigma_k^2 = \frac{\sum_{i=1}^{L} \xi_{k,i} (y'_i - \mu_k)^2}{\sum_{i=1}^{L} \xi_{k,i}}. \tag{12}$$

The above E-step and M-step are iteratively performed until $|L(\boldsymbol{y'}, \boldsymbol{\Theta}^{(t+1)}) - L(\boldsymbol{y'}, \boldsymbol{\Theta}^{(t)})| < T_L$ for an appropriately designed threshold T_L. The EM algorithm is known to converge in finite iterations for an arbitrary T_L.

An important property of the EM algorithm is that it is not guaranteed to converge to the global optimum. Instead, it stops at some local optimums, which can be much worse than the global optimum. In our model, the following constraints on the above parameters improve the accuracy of the performance. At least, we have two values $\hat{y}_i = \pm 1$ under the our attack model, and hence, we fix

$$\mu_1 = 1, \tag{13}$$

$$\mu_2 = -1. \tag{14}$$

All variances σ_k^2 are equal because $\hat{y}_i$ is distorted only by Gaussian noise.

If the “average” or “average2” attack is performed, the number of Gaussian components is at most $m = 3$; otherwise, $m = 2$ for collusion strategies under the marking assumption. When $m = 3$, the EM algorithm must estimate the following five parameters: a_1, a_2, a_3, μ_3 and $\sigma_e^2 (= \sigma_1^2 = \sigma_2^2 = \sigma_3^2)$. On the other hand, among these five parameters, a_3 and μ_3 are omitted when $m = 2$. Hence, the accuracy of the estimation at $m = 2$ is much better because the number of unknown parameters is reduced. Thus, the accurate estimation of m

will further improve the performance of EM algorithm when the number m is properly estimated.

For the estimation of m, we need to find the collusion strategy selected for producing a pirated copy. In [4], the EM algorithm is applied for the estimation of the collusion strategy. However, the experimental results indicate that the accuracy of the estimation is getting worse for more colluders and/or more harmful process. In our case, even if we wrongly estimate $m = 3$, the estimated parameters are not always bad. For example, when $a_3 = 0$ or $\mu_3 = 0$ in the case $m = 3$, the other parameters will be coincident with the case $m = 2$. So, we roughly determine m as follows:

$$m = \begin{cases} 2 & \text{if } \lambda(\boldsymbol{y'}) \geq L/2 \\ 3 & \text{otherwise,} \end{cases} \tag{15}$$

where $\lambda(\boldsymbol{y'})$ is the number of elements satisfying $|y'_i| \geq 1$.

3.2 Correlation Score

Suppose that we transmit over a Gaussian channel with input $\hat{\boldsymbol{y}}$ and output $\boldsymbol{y'}$. Now, the probability distribution function is given by Eq.(5). Here, we start with the case $m = 2$. Then,

$$\Pr[\hat{y}_i = 1 | y'_i] = \frac{a_1 N(y'_i; \mu_1, \sigma_e^2)}{a_1 N(y'_i; \mu_1, \sigma_e^2) + a_2 N(y'_i; \mu_2, \sigma_e^2)}, \tag{16}$$

and

$$\Pr[\hat{y}_i = -1 | y'_i] = \frac{a_2 N(y'_i; \mu_2, \sigma_e^2)}{a_1 N(y'_i; \mu_1, \sigma_e^2) + a_2 N(y'_i; \mu_2, \sigma_e^2)}. \tag{17}$$

In a noiseless case, we get $y'_i = \hat{y}_i$, and the correlation score $S_i^{(j)}$ is calculated by Eq.(2). Considering the above probabilities in a noisy case, Eq.(2) is rewritten by

$$\begin{aligned} S_i^{(j)} &= 1 \cdot \Pr[\hat{y}_i = 1 | y'_i] U_{j,i} + (-1) \cdot \Pr[\hat{y}_i = -1 | y'_i] U_{j,i}, \qquad (18) \\ &= \frac{a_1 N(y'_i; \mu_1, \sigma_e^2) - a_2 N(y'_i; \mu_2, \sigma_e^2)}{a_1 N(y'_i; \mu_1, \sigma_e^2) + a_2 N(y'_i; \mu_2, \sigma_e^2)} U_{j,i}. \qquad (19) \end{aligned}$$

Next, we generalize the above discussion. Now, we get the following probabilities:

$$\Pr[\hat{y}_i = 1 | y'_i] = \frac{a_1 N(y'_i; \mu_1, \sigma_e^2)}{\displaystyle\sum_{k=1}^{m} a_k N(y'_i; \mu_k, \sigma_e^2)}, \tag{20}$$

and

$$\Pr[\hat{y}_i = -1 | y'_i] = \frac{a_2 N(y'_i; \mu_2, \sigma_e^2)}{\displaystyle\sum_{k=1}^{m} a_k N(y'_i; \mu_k, \sigma_e^2)}. \tag{21}$$

Therefore, the correlation score $S_i^{(j)}$ is generally represented by

$$S_i^{(j)} = \frac{a_1 N(y_i'; \mu_1, \sigma_e^2) - a_2 N(y_i'; \mu_2, \sigma_e^2)}{\sum_{k=1}^{m} a_k N(y_i'; \mu_k, \sigma_e^2)} U_{j,i}. \tag{22}$$

3.3 Threshold

A simple approach to estimate the false-positive probability is to perform the Monte Carlo simulation. Indeed, it is not easy in general because of the heavy computational costs for estimating a tiny probability. Furon et al. proposed an efficient method estimating the probability of rare events [5]. The method can estimate the false-positive probability ϵ_1 for a given threshold Z, which means that the method calculates the map $\epsilon_1 = F(Z)$. Once the relations are obtained, it is sufficient to store them as a reference table. In other word, this method must be iteratively performed to obtain an objective threshold for a given ϵ_1.

In [7], an easy method to obtain a threshold for a given ϵ_1 has been proposed. The method is based on the CLT. At first, it calculates the variance of the correlation sum $S^{(\tilde{j})}$ such that an $\tilde{j}$-th codeword is randomly generated one and is not assigned to any user in a fingerprinting system. For a sufficient number of $\tilde{j}$, the variance σ_S^2 of $S^{(\tilde{j})}$ is calculated by $\sum(S^{(\tilde{j})} - E[S^{(\tilde{j})}])^2$, where $E[x]$ is the expectation of x. Because of the Gaussian approximation based on the CLT, the threshold Z for a given ϵ_1 can be calculated as follows:

$$Z = \sqrt{2\sigma_S^2} \cdot \mathrm{erfc}^{-1}\left(2\epsilon_1\right). \tag{23}$$

The disadvantage of this method is the uncertainty-based approximation because there is an argument about the validity of CLT applying for the estimation of ϵ_1.

Our main interest in this paper is to evaluate the traceability of the proposed detector compared with the conventional one. So, we roughly calculate the threshold Z by Eq.(23) for a given ϵ_1, and then, derive $F(Z)$ as the actual false-positive probability.

4 Equalization of Probability

Because of the symmetry of the bias distribution $f(p)$, it is expected to be $\Pr[\hat{y}_i = 1] = \Pr[\hat{y}_i = -1]$ unless the colluders do not know the actual values $X_{j,i}$ of their codewords. However, when they happen to get the values contained in segments, they can perform more active collusion strategies such as "all-0" and "all-1". Such a scenario is defined in [3] as the cryptographic colluders. Then, $\Pr[\hat{y}_i = 1]$ is not always equal to $\Pr[\hat{y}_i = -1]$. Under this condition, we reconsider the optimality of the proposed detector.

If the parameters a_1 and a_2 are accurately estimated by the EM algorithm,

$$\Pr[\hat{y}_i = 1] = a_1, \tag{24}$$

and

$$\Pr[\hat{y}_i = -1] = a_2. \tag{25}$$

Because of the imbalance between $\Pr[\hat{y}_i = 1]$ and $\Pr[\hat{y}_i = -1]$, it occurs the bias between the first term $\Pr[\hat{y}_i = 1|y'_i]U_{j,i}$ and the second term $\Pr[\hat{y}_i = -1|y'_i]U_{j,i}$ in Eq.(22). In order to equalize the bias of these probabilities, the correlation score $S_i^{(j)}$ is modified as follows:

$$\begin{aligned} S_i^{(j)} &= 1 \cdot \frac{\Pr[\hat{y}_i = 1|y'_i]}{\Pr[\hat{y}_i = 1]} U_{j,i} + (-1) \frac{\Pr[\hat{y}_i = -1|y'_i]}{\Pr[\hat{y}_i = -1]} U_{j,i}, \\ &= \frac{N(y'_i; \mu_1, \sigma_e^2) - N(y'_i; \mu_2, \sigma_e^2)}{\displaystyle\sum_{k=1}^{m} a_k N(y'_i; \mu_k, \sigma_e^2)} U_{j,i}. \end{aligned} \tag{26}$$

This modification also changes the distribution of the correlation sum S_j, and hence, the corresponding threshold must be accommodated. Thanks to the method in Sect.3.3, it is easy to derive the threshold Z under the above conversion of $S_i^{(j)}$.

5 Experimental Results

For the comparison of the performance of proposed methods, the number of detected colluders and the false-positive probability are evaluated for the Nuida code under the following conditions. The length is $L = 5000$, the number of users is $N = 10^4$ and the false-positive probability is $\epsilon_1 = 10^{-8}$. Under this condition, the total false-positive probability η is approximated to be 10^{-4}. In our attack model, a pirated codeword is produced by collusion attack using randomly selected 10^5 combinations of $\tilde{c} = 8$ colluders and it is distorted by additive white Gaussian noise. The performance of the tracing algorithms is evaluated by changing SNR. Using a threshold Z calculated by Eq.(23), η is evaluated by $F(Z)$ as well as the Monte Carlo simulation. We denote the detector proposed in Sect.3 and Sect.4 by "method I" and "method Ⅱ", respectively. The threshold for the EM algorithm is set to be $T_L = 0.01$. In order to reduce the computational costs required for each trial of a Monte Carlo simulation, the number of iterations for the EM algorithm is limited to be 100 at most.

The number of detectable colluders under the "majority" attack is plotted in Fig.1. It is observed that both of the proposed methods approach to that of SD method in the decrease of SNR, and that the method Ⅱ outperforms the other methods. The reason why the traceability of method I is dropping with the increase of SNR comes from the wrong estimation of parameters in the EM algorithm. Such a wrong estimation is occurred in the case that the estimator judges $m = 3$ when in fact $m = 2$. By intensively measuring the estimated values, we found that μ_3 is very close to one of μ_1 and μ_2 in many cases. It means that the EM algorithm finds only two distribution in spite of the wrong judgment of $m = 3$. In case $\mu_3 \approx 1(= \mu_1)$, we see $\Pr[\hat{y}_i = 1] = a_1 + a_3$, but it is judged $\Pr[\hat{y}_i = 1] = a_1$ by mistake in the proposed method I, which affects on the probability $\Pr[\hat{y}_i = 1|y'_i]$. As the result, the score $S_i^{(j)}$ given by Eq.(22) is affected by the miscalculation in the method I. By contrast, the score $S_i^{(j)}$ in Eq.(26) in the method II is stable for the miscalculation. Assuming an ideal case that the EM algorithm can estimate the parameters with no error, the performance of the proposed methods is evaluated under a same condition. For the comparison, we plot the results of ideal case by solid lines and the actual values by dotted lines in Fig.2. We can see that the traceability of method I is very close to, but is slightly lower than that of method II in an ideal case. For further comparison, we check the performance in the ideal case under the other collusion attacks for 10^3 trials of Monte Carlo simulation, which results are described in Fig.3. Notice that the results of method II under "all-0" and "all-1" collusion strategies are much higher than that of method I. It comes from the effect of equalization explained in Sect.4. From this result, we can say that colluders can not get any benefit from the information of symbols embedded in a copy. Under the "WCA", we also evaluate the performance for 10^5 trials of Monte Carlo simulation, which results are plotted in Fig.4. The results are almost equal to those of the "majority" attack.

Even if the score of innocent users can be approximated by a Gaussian distribution, the probability of false-positive cannot be simply expressed by Gauss error function. The total false-positive probabilities under the "majority" attack and "WCA" are plotted in Fig.5. In these figures, the solid and dotted lines are the results derived from the experiment and $F(Z)$, respectively. Although the experimental results are slightly dispersed because the number of Monte Carlo simulation is only 10^5, they are almost equal to $F(Z)$ and are less than a given probability $\eta = 10^{-4}$. It means that the Gaussian approximation based on the CLT for calculating the threshold Z is not bad under this condition.

In order to numerically compare the performance against collusion strategies, the number of detected colluders and the total false-positive probability are summarized in Table 1 and Table 2, respectively. As a whole, it is observed that the traceability of the method II is better than that of the method I, and the method II outperforms the conventional methods. It is remarkable that the total false-positive probability of "minority" attack is the worst one among 8 collusion strategies under this experimental condition. Since our scope in this paper is not to evaluate the validity of Gaussian assumption, but to calculate a proper correlation score $S_i^{(j)}$ under the noisy environment, the design of appropriate threshold Z is not deeply discussed and we merely employ the Gaussian assumption to calculate Z for a given ϵ_1 for its simplicity. Indeed, the use of rare event simulator $F(Z)$ can be a better method for designing the threshold though it requires an iterative search for obtaining an objective threshold for a given ϵ_1.

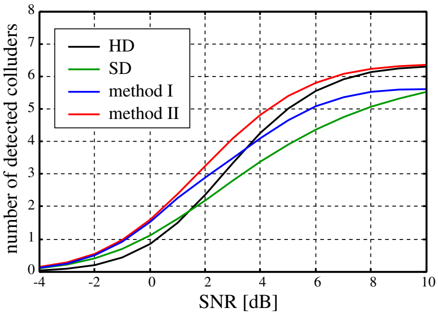


Fig. 1. Comparison of the traceability under the majority attack for $L = 5000$, $\tilde{c} = 8$, and $\epsilon_1 = 10^{-8}$

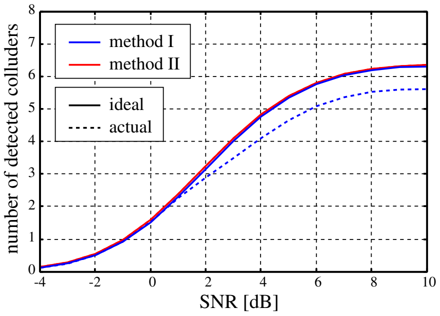


Fig. 2. Comparison of the traceability of ideal case under the majority attack for $L = 5000$, $\tilde{c} = 8$, and $\epsilon_1 = 10^{-8}$

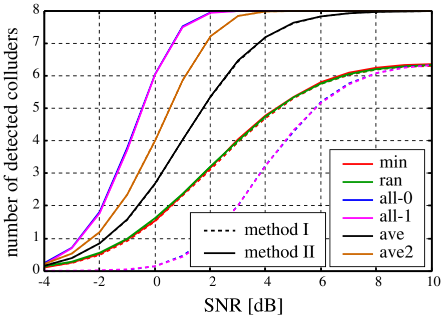


Fig. 3. Comparison of the traceability under various collusion strategies for $L = 5000$, $\tilde{c} = 8$, and $\epsilon_1 = 10^{-8}$

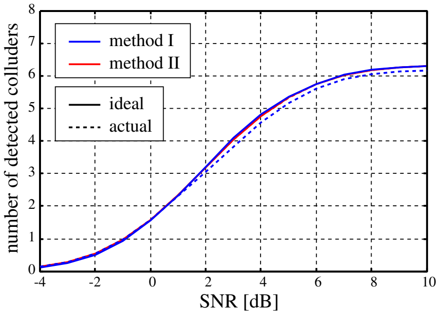
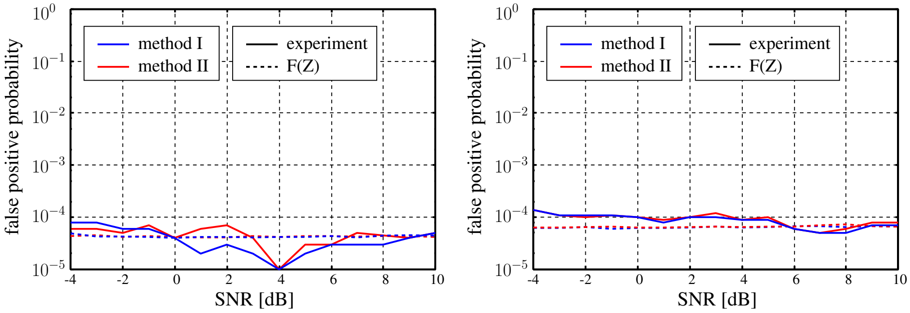


Fig. 4. Comparison of the traceability under the WCA for $L = 5000$, $\tilde{c} = 8$, and $\epsilon_1 = 10^{-8}$



(a) majority (b) WCA

Fig. 5. Comparison of the total false-positive probability η for $L = 5000$, $\tilde{c} = 8$, and $\epsilon_1 = 10^{-8}$

Table 1. Number of detected colluders for 8 collusion strategies when $L = 5000$ and $\tilde{c} = 8$

SNR	detector	collusion strategy							
[dB]		maj	min	ran	all-0	all-1	ave	ave2	WCA
−4	HD	0.037	0.042	0.040	0.040	0.039	0.038	0.040	0.039
	SD	0.109	0.123	0.117	0.118	0.116	0.187	0.243	0.113
	method I	0.120	0.132	0.137	0.001	0.001	0.179	0.202	0.134
	method II	0.132	0.145	0.137	0.262	0.260	0.179	0.202	0.135
0	HD	0.860	0.881	0.874	0.878	0.865	0.864	0.872	0.868
	SD	1.125	1.132	1.132	1.157	1.140	2.779	4.058	1.128
	method I	1.532	1.545	1.579	0.172	0.165	2.586	3.872	1.574
	method II	1.596	1.608	1.579	6.079	6.069	2.780	4.060	1.574
4	HD	4.270	4.242	4.257	4.268	4.241	4.258	4.250	4.255
	SD	3.425	3.391	3.410	3.471	3.440	7.240	7.969	3.413
	method I	4.109	4.107	4.547	5.418	5.394	7.132	7.984	4.562
	method II	4.822	4.817	4.768	8.000	8.000	7.138	7.985	4.762
8	HD	6.144	6.131	6.143	6.148	6.131	6.139	6.135	6.137
	SD	5.097	5.078	5.098	5.160	5.127	7.972	8.000	5.097
	method I	5.604	5.531	6.041	7.016	7.006	7.547	8.000	6.054
	method II	6.228	6.232	6.183	8.000	8.000	7.966	8.000	6.174

Table 2. False-positive probability $\eta[\times 10^{-4}]$ experimentally derived for 8 collusion strategies when $L = 5000$ and $\tilde{c} = 8$, where the values in parenthesis are $F(Z)$

SNR	detector	collusion strategy							
[dB]		maj	min	ran	all-0	all-1	ave	ave2	WCA
−4	method I	0.8	1.5	0.3	0.7	0.9	0.1	0.3	1.4
		(0.432)	(1.183)	(0.734)	(0.227)	(0.233)	(0.504)	(0.720)	(0.620)
	method II	0.6	1.3	0.3	0.4	0.3	0.1	0.3	1.4
		(0.432)	(1.183)	(0.734)	(0.227)	(0.233)	(0.504)	(0.720)	(0.620)
0	method I	0.4	0.8	0.4	0.7	1.4	0.2	0.3	1.0
		(0.411)	(1.202)	(0.693)	(0.073)	(0.077)	(0.252)	(0.347)	(0.628)
	method II	0.4	1.0	0.5	0.1	0.0	0.2	0.3	1.0
		(0.403)	(1.202)	(0.693)	(0.073)	(0.077)	(0.252)	(0.347)	(0.628)
4	method I	0.1	1.6	0.6	0.6	1.2	0.2	0.1	0.9
		(0.414)	(1.215)	(0.716)	(0.039)	(0.038)	(0.245)	(0.148)	(0.636)
	method II	0.1	1.2	1.0	0.0	0.3	0.1	0.1	0.9
		(0.414)	(1.212)	(7.416)	(0.038)	(0.038)	(0.245)	(0.148)	(0.636)
8	method I	0.3	1.2	0.7	0.1	1.0	0.0	0.0	0.5
		(0.431)	(1.221)	(0.699)	(0.029)	(0.032)	(0.035)	(0.079)	(0.644)
	method II	0.4	1.1	0.7	0.0	0.2	0.0	0.0	0.6
		(0.431)	(1.221)	(0.699)	(0.029)	(0.032)	(0.035)	(0.079)	(0.644)

6 Conclusion

In this paper, we proposed a soft decision tracing algorithm to catch more colluder even if a pirated codeword is distorted by Gaussian noise. We first estimate the parameters of Gaussian channel using the EM algorithm by giving some constrains. Then, the correlation score is calculated using the posterior probability of each symbol of received codeword. Considering the bias between the probability of symbols, we give a weight on the posterior probability. The experimental results show that the proposed method without the weighting requires an accurate estimation of the number of Gaussian mixture model to get a best performance, and the method with the weighting is not so sensitive for such an estimation. For the specific collusion strategies such as “all-0” and “all-1”, it is confirmed from our experiment that the weighting effectively enhances the performance of tracing algorithm.

Although the proposed method is specified for AWGN channel, it can be extended for further complicated attack channels by tuning the EM algorithm. For example, if additive *colored* Gaussian noise is injected to a pirated codeword, we must estimate the mean values μ_1 and μ_2, while they are fixed under the AWGN channel. Furthermore, when the distribution of additive noise is modeled by a certain distribution such as Laplace and Rayleigh distributions, it is sufficient to replace the Gaussian term $N(y'_i; \mu, \sigma^2)$ appeared in this paper with the modeled one.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
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: Solanki, K., Sullivan, K., Madhow, U. (eds.) IH 2008. LNCS, vol. 5284, pp. 341–356. Springer, Heidelberg (2008)
4. Furon, T., Preire, L.P.: EM decoding of Tardos traitor tracing codes. ACM Multimedia and Security, 99–106 (2009)
5. Furon, T., Preire, L. P., Guyader, A., Cérou, F.: Estimating the minimal length of Tardos code. In: Katzenbeisser, S., Sadeghi, A.-R. (eds.) IH 2009. LNCS, vol. 5806, pp. 176–190. Springer, Heidelberg (2009)
6. Kuribayashi, M.: Experimental assessment of probabilistic fingerprinting codes over AWGN channel. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) IWSEC 2010. LNCS, vol. 6434, pp. 117–132. Springer, Heidelberg (2010)
7. Kuribayashi, M.: Tardos's fingerprinting code over AWGN channel. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 103–117. Springer, Heidelberg (2010)
8. Kuribayashi, M., Morii, M.: Systematic generation of Tardos's fingerprinting codes. IEICE Trans. Fundamentals E93-A(2), 508–515 (2009)
9. Nuida, K.: Making collusion-secure codes (more) robust against bit erasure. eprint 549, 2009 (2009)

10. 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)
11. Nuida, K., Hagiwara, M., Watanabe, H., Imai, H.: Optimization of Tardos's fingerprinting codes in a viewpoint of memory amount. In: Furon, T., Cayre, F., Doërr, G., Bas, P. (eds.) IH 2007. LNCS, vol. 4567, pp. 279–293. Springer, Heidelberg (2008)
12. Simone, A., Skoric, B.: Accusation probabilities in Tardos codes: the Gaussian approximation is better than we thought. Cryptology ePrint Archive, Report 2010/472 (2010), http://eprint.iacr.org/
13. Tardos, G.: Optimal probabilistic fingerprint codes. J. ACM 55(2), 1–24 (2008)
14. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion resistant fingerprinting for multimedia. IEEE Signal Processing Mag., 15–27 (2004)