Joint Coding and Embedding Techniques for Multimedia Fingerprinting

Shan He, *Student Member, IEEE*, and Min Wu, *Member, IEEE*

***Abstract*—Digital fingerprinting protects multimedia content from illegal redistribution by uniquely marking every copy of the content distributed to each user. The collusion attack is a powerful attack where several different fingerprinted copies of the same content are combined together to attenuate or even remove the fingerprints. One major category of collusion-resistant fingerprinting employs an explicit step of coding. Most existing works on coded fingerprinting mainly focus on the code-level issues and treat the embedding issues through abstract assumptions without examining the overall performance. In this paper, we jointly consider the coding and embedding issues for coded fingerprinting systems and examine their performance in terms of collusion resistance, detection computational complexity, and distribution efficiency. Our studies show that coded fingerprinting has efficient detection but rather low collusion resistance. Taking advantage of joint coding and embedding, we propose a permuted subsegment embedding technique and a group-based joint coding and embedding technique to improve the collusion resistance of coded fingerprinting while maintaining its efficient detection. Experimental results show that the number of colluders that the proposed methods can resist is more than three times as many as that of the conventional coded fingerprinting approaches.**

***Index Terms*—Collusion resistance, error correcting code, group-based fingerprinting, joint coding and embedding, multimedia fingerprinting, traitor tracing.**

I. Introduction

TECHNOLOGY advancement has made multimedia content widely available and easy to process. These benefits also bring ease to unauthorized users who can duplicate and manipulate multimedia content, and redistribute it to a large audience. As such, the protection of multimedia content becomes increasingly important. Digital fingerprinting is an emerging technology to protect multimedia content from unauthorized dissemination, where each user's copy is identified by a unique ID, known as a fingerprint, embedded in his or her copy and the fingerprint can be extracted to help identify culprits when a suspicious copy is found. A powerful, cost-effective attack from a group of users is collusion, where several users combine their copies of the same content to generate a new version. If designed improperly, the fingerprints can be weakened or removed by a collusion attack.

A growing number of techniques have been proposed recently concerning collusion-resistant fingerprinting for multimedia. Many of them fall in one of two categories, according to whether an explicit discrete coding step is involved. In the noncoded category, a typical example is orthogonal fingerprinting, which assigns each user a spread-spectrum (SS) sequence as a fingerprint, and the sequence is typically orthogonal to those for other users [1], [2]. The collusion resistance performance of orthogonal fingerprinting can be improved by introducing correlation to the fingerprints for users who are likely to collude together due to cultural and other relations [3]. Noncoded fingerprinting is a natural extension from SS embedding [4] and is easy to implement. A weakness of noncoded schemes is that the required number of spreading sequences and the computational complexity of detection would increase linearly with the number of users.

Building coded fingerprints for generic data (such as executable software programs and bitstreams) was investigated by the coding and cryptography communities. Early works can be traced back to the 1980s [5], [6]. A concept of marking assumption was introduced by Boneh and Shaw in [7], and a two-level binary code construction, known as a c-secure code, was proposed to resist up to c colluders with high probability. This binary code was later used to modulate a direct SS sequence to embed fingerprint codes into multimedia signals [8]. By explicitly exploiting the multimedia characteristics through selecting appropriate modulation and embedding schemes, a more compact code was introduced in [9] based on combinatorial design to identify colluders through the code bits shared by them. Many recent works on coded fingerprinting [10], [11] extend Boneh and Shaw's framework and consider the construction of codes with traceability, such as the identifiable parent property (IPP) code and the traceability (TA) code. Among these codes, TA codes are stronger than other codes in terms of tracing capability and can be systematically constructed using well-established error correcting code (ECC). Thus, TA codes are widely used in the coded fingerprinting literature. The authors of [12] and [13] applied the ECC-based TA code to multimedia fingerprinting and extended it to deal with symbol erasures contributed by noise or cropping in the multimedia signal domain. Another reason why researchers favor ECC for fingerprint code construction is that some ECCs, such as the algebraic-geometry

Manuscript received April 30, 2005; revised February 4, 2006. This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N00014-05-10634 and in part by the U.S. National Science Foundation under CAREER Award CCR-0133704. Preliminary results of this work were presented in SPIE Conference on Security, Watermarking and Stegonography, San Jose, 2005 [26] and IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, 2005 [24]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jerry Gibson.

The authors are with the Department of Electrical and Computer Engineering and the Institute of Advanced Computing Studies, University of Maryland, College Park, MD 20742 USA (e-mail: shanhe@eng.umd.edu; minwu@eng.umd.edu).

Digital Object Identifier 10.1109/TIFS.2006.873597

codes, have efficient decoding algorithms. For example, the authors in [14] employed the Guruswami–Sudan soft-decision list decoding algorithm for the algebraic-geometry code to identify multiple colluders. In this paper, we focus on the coded fingerprinting constructed by ECC and refer to it as the ECC-based fingerprinting.

In the existing coded fingerprinting works that originated from fingerprinting generic data, the special properties and issues of the multimedia signal have not been sufficiently explored in the code design. Although some papers [12], [14] claimed that their schemes are for multimedia, the embedding issues are handled in a rather abstract level through models based on the marking assumptions. They typically assume that colluders can only change fingerprint symbols in which they have different values, and that the colluders assemble pieces of their codewords to generate a colluded version. Although the marking assumptions may work well with generic data, they alone are not capable of modeling multimedia fingerprinting, where colluders can manipulate fingerprinted multimedia in the signal domain to bring code-domain changes beyond the marking assumptions. In the meantime, as has been shown in [9], by jointly exploring embedding and coding, we can substantially limit the effective ways that attackers may exploit; for example, they cannot manipulate the bits/symbols on the code level. Thus, it is important to examine the overall performance across coding and signal domains, taking into account the coding, embedding, attack, and detection issues.

In this paper, we start with introducing a general framework for coded multimedia fingerprinting by integrating coding and embedding issues. Focusing on ECC code construction, we examine the overall performance of ECC-based multimedia fingerprinting across both coding and embedding layers. As will be shown in the paper, the ECC-based fingerprinting has more efficient detection in terms of computational complexity than noncoded orthogonal fingerprinting, but its colluder traceability is considerably lower. In order to achieve a better tradeoff between the collusion resistance and detection computational complexity, we jointly consider coding and embedding during fingerprint design. First, we observe a huge gap between the resistance of coded fingerprinting against different collusion attacks and, particularly, interleaving collusion is much more effective than averaging collusion from the attackers' perspective. We thus propose a permuted subsegment embedding technique to enforce interleaving collusion to have a similar effect on the embedded fingerprints to what averaging collusion brings. The key idea is to divide each segment of the fingerprint, which corresponds to one symbol, into several subsegments, and then to randomly permute these subsegments before embedding. At the detection stage, inverse permutation is performed on these subsegments, followed by a correlation detector to identify traitors. Second, taking advantage of prior knowledge that some users are more likely to collude together than with others, possibly due to geographical or cultural reasons [3], we propose a group-based joint coding and embedding (GRACE) technique. In GRACE, each fingerprint consists of a user subcode and a group subcode, and is embedded in the host signal via the SS technique. The detection is done in two levels, which identifies guilty groups through correlation and then narrows down to specific colluders through minimum distance decoding or correlation-based soft detection. The comparison between the proposed fingerprinting schemes and the existing ECC-based fingerprinting shows that the fingerprinting strategy of joint coding and embedding substantially improves the collusion resistance of ECC-based fingerprinting, while preserving its advantages of compact representation and efficient detection.

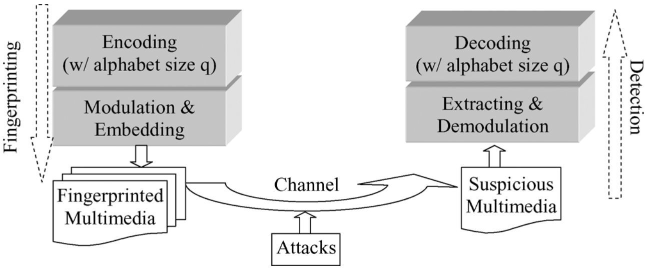


Fig. 1. Framework of the embedded ECC-based fingerprinting.

The paper is organized as follows. Section II provides a general background on ECC-based fingerprinting. Section III examines the detection efficiency and collusion traceability of the conventional ECC-based fingerprinting. Based on the results obtained from Section III, we propose the permuted subsegment embedding technique in Section IV and show its effectiveness through experimental results. We present in Section V the proposed GRACE technique, along with the design and performance evaluation of multimedia fingerprinting systems integrating the two proposed techniques. Finally, conclusions are drawn in Section VI.

II. Background on ECC-Based Fingerprinting

Fingerprint construction and embedding are two important issues for a multimedia fingerprinting system. We illustrate a framework of applying coded fingerprinting for multimedia data in Fig. 1, which consists of a coding layer and an embedding layer. In fingerprinting applications, the original host signal is typically available to detectors [15], which is known as nonblind detection, and the robustness against a single user's attacks (such as noise addition, compression, and filtering) is a basic requirement. The SS additive embedding technique or its variations is a viable choice for the embedding layer, owing to its excellent robustness under nonblind detection that has been demonstrated in the literature [4]. A symbol in a fingerprint code over an alphabet of size q can be mapped to a signal suitable for embedding through various modulation techniques [16]. Orthogonal modulation that uses q mutually orthogonal signals to represent q symbol values widely separates the different symbols in the signal domain and, thus, gives higher detection accuracy.

The prior works on ECC-based fingerprinting have been designed on top of the marking assumptions [12], [17]. We now replace the abstraction of marking assumptions with a modulation and embedding layer for a complete system of multimedia fingerprinting. Thus, the layered structure of the ECC-based fingerprinting system includes an ECC code layer and an SS-based embedding layer, along with an attack channel where we mainly focus on collusion attacks. In the following, we shall address several important issues of ECC-based fingerprinting over the three main stages, namely fingerprinting, collusion attacks, and detection.

A. Fingerprinting

During the fingerprinting process, we first choose an ECC code over an alphabet with size q, and assign a codeword to each user. The design requirement of this ECC fingerprint code will be discussed later in this section.

We partition the host signal into nonoverlapped segments, where each segment is to carry one symbol of the fingerprint code. The partition can be done spatially into blocks for image, or temporally into frames for video and audio. Within each segment, we use q mutually orthogonal SS sequences $\{\mathbf{u}_i, i = 1, \ldots, q\}$ with identical energy $||\mathbf{u}||^2$ to represent the q possible symbol values, and add one of these sequences into the segment (with perceptual scaling) according to the symbol value in the fingerprint code. Each fingerprinted segment can be modeled as

$$\mathbf{y}_{jk} = \mathbf{u}_{\mathrm{sym}(j,k)} + \mathbf{x}_k \tag{1}$$

where $\mathbf{x}_k$ is the kth segment of a host signal, and $\mathbf{y}_{jk}$ is the kth fingerprinted segment for the jth user. The function $\mathrm{sym}(j,k)$ is used to retrieve the symbol for the kth segment from the jth user's codeword, and $\mathbf{u}_{\mathrm{sym}(j,k)}$ is the SS sequence corresponding to the symbol value. The concatenation of all fingerprinted segments forms the ultimate fingerprinted signal.

B. Collusion Attacks

In most existing works concerning fingerprinting, it is assumed that the colluders can only change the fingerprint code symbols where they see different values within the colluder group [7], and a colluded version is constructed by assembling pieces of the colluders' codewords [12]. We refer to this as (symbol-based) interleaving collusion. Additional distortion may be added to the multimedia signal during the collusion, which we model as additive noise. Since few colluders would be willing to take higher risk than others, they generally would make contributions of an approximately equal amount in the collusion [15].

In addition to interleaving collusion, colluders can manipulate fingerprinted multimedia in the signal domain, incurring a variety of code-domain changes beyond the marking assumptions. A simple, yet effective way is to average the corresponding signal components or features from multiple copies [9], bringing changes that are different from interleaving collusion. The averaging collusion can be modeled as follows:

$$\mathbf{z} = \frac{1}{c}\sum_{j \in S_c} \mathbf{s_j} + \mathbf{x} + \mathbf{d} \tag{2}$$

where $\mathbf{z}$ is the colluded signal, $\mathbf{x}$ is the host signal, $\mathbf{d}$ is the noise term, $\mathbf{s}_j$ represents the fingerprint sequence for user j, S_c is the colluder set, and c is the number of colluders. Studies in [18] have shown that a number of nonlinear collusions can be well approximated by an averaging collusion plus additive noise. Thus, we will mainly focus on the interleaving and averaging collusions in this paper. For simplicity in analysis, we assume that the additional noise under both collusions follows independently identically distributed (i.i.d.) Gaussian distribution. The effects of many other distortions have been studied in the watermarking literature, such as quantization/compression and geometric distortions. And since the original host signal is often available to detector in fingerprinting applications, we can use it as a reference and the effects of many distortions can be approximated well by additive noise.

C. Detection

At the detector side, our goal is to catch one of the colluders with high probability. We first determine which symbol is present in each multimedia segment through a correlation detector commonly used for SS embedding [2], [4]. As a host signal can be made available to detectors in many fingerprinting applications, we register the suspicious copy with the host signal and subtract the host signal from the suspicious copy to obtain a test signal. Then, for each segment of the test signal, we employ a maximum correlation detector to identify the symbol; that is, we correlate it with each of the q spreading sequences, identify the sequence giving the maximum correlation, and record the corresponding symbol. The detection statistic for the kth segment is defined as

$$T_s(k,i) = \frac{(\mathbf{z}_k - \mathbf{x}_k)^T \mathbf{u}_i}{\sqrt{||\mathbf{u}_i||^2}}, \quad i = 1, 2, \ldots, q \tag{3}$$

where $\mathbf{z}_k$ and $\mathbf{x}_k$ represent the kth segment of the colluded signal and that of original signal, respectively. The extracted symbol from the kth segment is $\hat{i} = arg\max_{i=1,2,\ldots,q} T_s(k,i)$. With the sequence of symbols extracted from all segments using this maximum detector, we proceed to the ECC code layer and apply a decoding algorithm to identify the colluder whose codeword has the most matched symbols with the extracted symbol sequence.

Alternatively, we can employ a soft-detection strategy to keep the correlation results of (3) with each of the q possible sequences at every segment without determining the symbol value, and then collect the results from all segments together to arrive at the correlation result for each user as

$$T_N(j) = \sum_{k=1}^{L} T_s\left(k, \mathrm{sym}(j,k)\right) \quad j = 1, 2, \ldots, N_u \tag{4}$$

where L is the code length, and N_u is the total number of users. Note that this approach has the correlation results equivalent to[1]a matched-filter detector that correlates the entire test signal with each user's fingerprint sequence $\mathbf{s}_j$ by

$$T_N(j) = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{s}_j}{\sqrt{||\mathbf{s}||^2}} \quad j = 1, 2, \ldots, N_u. \tag{5}$$

Here, $||\boldsymbol{s}|| = ||\boldsymbol{s}_j||$ for all j based on the equal energy construction. The user whose fingerprint has the highest correlation value $T_N(j)$ is identified as the colluder (i.e.,

[1] As we shall see later in Section III-A, computing the partial correlation and then aggregating together is a more efficient implementation than taking the N_u correlation results on the whole signal. In this paper, we shall employ this efficient implementation for the matched-filter detector in (5) for ECC-based fingerprinting.

$\hat{j} = arg\max_{j=1,2,\ldots,N_u} T_N(j)$). Compared with the former two-step hard-decision scheme, the latter scheme takes advantage of the soft information on the symbol level and provides a better collusion identification performance. In both hard and soft detectors, we always make decisions on the colluder identification and only accuse one user as the colluder. Therefore, the probability of false positive will be one minus the probability of detection.

Under the above framework, the noncoded orthogonal fingerprinting can be seen as a special case that the alphabet size q equals the total number of users N_u and the codeword length equals 1. The detection for orthogonal fingerprinting is done by first correlating the test signal with each user's sequence and then identifying the user with the highest correlation statistic as the colluder.

D. Considerations on ECC Fingerprint Codes

A common practice in fingerprint code design treats the symbols contributed from other colluders as errors, and makes the minimum distance between codewords large enough to tolerate the errors. The minimum distance requirement ensures that the best match with a colluded codeword (referred to as the descendant) comes from one of the true colluders. The c-TA code [17] is such an example.

Let $\Gamma \subseteq Q^L$ be a code over an alphabet Q with length L and N_u codewords. Without loss of generality, we consider the first c users as colluders. The set of c colluders is denoted as $C = \{\mathbf{v}_1, \ldots, \mathbf{v}_c\} \subset \Gamma$, where a codeword $\mathbf{v}_i \in \Gamma$ represents the ith colluder and consists of a sequence of L symbols (i.e., $\mathbf{v}_i = [w_1^{(i)} w_2^{(i)} \ldots w_L^{(i)}]$). A codeword set that can descend from this colluder set is denoted as

$$\mathrm{desc}(C) = \left\{ [x_1 \ldots x_L] : x_j \in \left\{ w_j^{(i)} : 1 \le i \le c \right\}, 1 \le j \le L \right\}.$$

If for any descendant $[x_1 \ldots x_L] \in \mathrm{desc}(C)$, there is a $\mathbf{v}_i \in C$ such that

$$\left| \left\{ j : x_j = w_j^{(i)} \right\} \right| > |\{ j : x_j = s_j \}|$$

for any innocent user's codeword $[s_1 \ldots s_L] \in \Gamma \setminus C$, where the notation $|\cdot|$ is the cardinality, then Γ is called a c-traceability (c-TA) code and denoted as $c - \mathrm{TA}_q(L, N_u)$ with $q = |Q|$.

Under the conventional marking assumptions, a c-TA code can be constructed using an ECC if its minimum distance D satisfies [17]

$$D > \left(1 - \frac{1}{c^2}\right) L \tag{6}$$

where L is the code length and c is the colluder number.

As mentioned earlier, most of the existing works [12]–[14] mainly consider the outer layer of the system (i.e., the ECC code layer), and deal with the embedding through marking assumptions. However, the distortions and attacks mounted by adversaries on the fingerprinted multimedia can lead to errors in detecting fingerprint code symbols, which are beyond the marking assumptions. The existing work on c-TA codes has been extended to tolerate erasures [12]. Recently, we have further extended the work by considering both erasures and nonerasure errors [19].

As can be seen from the above discussions, the ECC-based fingerprint code prefers an ECC with the larger minimum distance to tolerate more colluders. Among ECC constructions, Reed–Solomon codes have the minimum distance that achieves the Singleton bound [20] and is widely used in the existing coded fingerprinting works [12], [17]. We employ a q-ary Reed–Solomon code with code length L to construct a c-TA code. The parameters of the L-tuple Reed–Solomon code for N_u users should satisfy [19]

$$N_u = q^t, \quad \text{and} \quad t = \left\lceil \frac{L}{c^2} - \frac{c+1}{c^2} L_{\mathrm{FA}} \right\rceil \tag{7}$$

where L_{FA} is an auxiliary parameter indicating the number of symbol errors the code is designed to tolerate.

In general, the decoding computational complexity of the c-TA code is $O(N_u)$ for a total of N_u codewords. For Reed–Solomon codes, or more generally algebraic-geometry codes, there is a more efficient decoding method known as the list decoding, which can correct more errors than the decoding radius imposed by the minimum distance. The list decoding algorithm can reduce the decoding complexity to the order of polynomial in $c \log N_u$ [21]. However, as we will see in the following section, when we take the embedding layer into consideration, the demodulation process to extract the embedded symbols dominates the accounting of the detection computational complexity. This also suggests the importance of the joint consideration of coding and embedding.

III. Performance Evaluation of ECC-Based Fingerprinting

Examining the existing literature on ECC-based fingerprinting reveals that few works actually considered the embedding of the designed fingerprints into a host signal and the extraction of them after the collusion. We have found a very limited amount of overall performance analysis by considering the coding and embedding together [8], and little comparison with noncoded orthogonal fingerprinting. Thus, in this section, we first analyze the computational complexity of the detection process and the efficient distribution of ECC-based fingerprinting. We then examine its collusion resistance through measuring the probability of catching one colluder under different values of the colluder number and compare it with the performance of noncoded orthogonal fingerprinting.

A. Computational Complexity of Detection

As we have pointed out in the previous section, one of the reasons that researchers in the literature may favor ECC-based fingerprinting over the noncoded orthogonal approach is because some classes of ECC have more efficient decoding algorithms than the maximum-likelihood decoding that is commonly used for orthogonal fingerprinting [22]. By jointly considering the coding and embedding of ECC-based fingerprinting, we can obtain a complete picture on the computational complexity for colluder identification, which consists of demodulation and decoding. We shall show that while the efficient decoding improves the detection efficiency, the improvement is a relatively small part in the overall computational complexity. The major improvement on the detection efficiency comes from the demodulation process.

For a fingerprinting system with a total of N_u users and a host signal with totally N embeddable components, the detection of orthogonal fingerprinting is done by correlating the test signal with each user's fingerprint sequence. This takes N_uN multiplications plus $N_u(N-1)$ summations, or a total of $O(N_uN)$ operations. We further perform N_u-1 comparisons to find the fingerprint sequence corresponding to the highest correlation to identify one of the colluders. Thus the computational complexity of the whole detection process is $O(N_uN)+O(N_u)=O(N_uN)$.

For ECC-based fingerprinting, since the fingerprint sequences for each segment only have q different versions (corresponding to q symbols), we only need $qL(N/L)$ multiplications plus $qL(N/L-1)$ summations and $L(q-1)$ comparisons for demodulation, giving a total computational complexity of $O(qN)$. In the decoding step, we can determine the colluder through N_uL+N_u-1 comparisons by brute force searching, which provides an upper bound on the decoding complexity. Putting the demodulation and decoding steps together, we find the computational complexity for ECC-based fingerprinting as $O(qN)+O(N_uL)$. In many practical applications of robust fingerprinting, to ensure fingerprints be reliably embedded in multimedia, we generally have $N_u \ll N$. This suggests that the demodulation part dominates the overall complexity, regardless of the use of efficient decoding algorithms. Therefore, the overall computational complexity becomes $O(qN)$. Similarly, the soft detector of (5) with implementation of (4) needs $O(qN)$ operations to calculate the partial correlations and further requires $O(N_uL)$ summations and N_u-1 comparisons to determine the colluder. This leads to the same computational complexity bound of $O(qN)$ as the hard detection. Taking a Reed–Solomon code construction with $N_u=q^t$ as an example, we obtain the bound of detection computational complexity for ECC-based fingerprinting as $O(\sqrt[t]{N_u}N)$.

Comparing the detection computational complexity of ECC-based fingerprinting and orthogonal fingerprinting, we can see that the significant improvement on the demodulation process brings a substantial advantage of ECC-based fingerprinting over the orthogonal fingerprinting. This is largely owing to the reduced alphabet size in ECC-based fingerprinting. Furthermore, we notice that ECC-based fingerprinting requires as few as q orthogonal sequences of length N/L, while the orthogonal fingerprinting requires N_u mutually orthogonal sequences of length N. This suggests that the ECC-based system has an advantage of providing a more compact way of representing users and consuming fewer resources in terms of the orthogonal sequences. The compact representation of fingerprints allows for a simpler design and implementation in the embedding and detection stages.

B. Efficient Distribution of Fingerprinted Signals

In some applications, such as video streaming, where a huge amount of data has to be transmitted to a number of users in real time, the efficient generation and distribution of fingerprinted copies for different users is an important issue. ECC-based fingerprinting provides a potential support for the efficient distribution of the fingerprinted signal. This is because for a total of N_u users, every segment only has q versions, each of which has one of the q possible symbols embedded. We can pregenerate these q versions for each segment, which allows us to quickly construct the fingerprinted copy for any given user by concatenating the corresponding segments according to his or her codeword. To distribute these fingerprinted copies, we can employ secure multicast protocols such as that by Chu *et al.* [23]. Since for each segment we send q copies, the bandwidth requirement on the sender side for distributing N_u copies is qB, where B is the bandwidth requirement of sending only one copy.

In contrast, for an orthogonal fingerprinting system, all users have different versions at each segment. There is no structural advantage we can take in constructing and distributing the fingerprinted signals. The owner needs to generate the whole fingerprinted signal for each user and to unicast one of the N_u versions of the signals to each user, which generally requires a bandwidth of N_uB.

We compare the communication cost of ECC-based fingerprinting and orthogonal fingerprinting by defining γ as the ratio of the bandwidth consumption of ECC-based fingerprinting to that of orthogonal fingerprinting. From the above discussion, we have $\gamma = qB/(q^tB) = q^{1-t}$. When the ECC-based fingerprinting is constructed based on a Reed–Solomon code, for example, with parameters $t=2$, $q=32$, γ has value of 1/32. This suggests that the communication bandwidth required by a sender employing ECC-based fingerprinting can be one to two orders of magnitude lower than that of orthogonal fingerprinting. If the communication cost requirement is more stringent than other parameters, we can further adjust t to lower the cost.

C. Analysis of Collusion Resistance

Consider an ECC-based fingerprinting system employing a L-tuple code with minimum distance D over q-ary alphabet to represent N_u users. Under the (symbol wise) interleaving collusion, the colluders exploit the fingerprint pattern and contribute segment by segment with each segment carrying one symbol. Averaging collusion does not rely on the fingerprint pattern and simply takes the average value of each signal component. As a result, these two collusion attacks have different effects on collusion detection and we shall analyze them separately.

1) Interleaving Collusion: During the interleaving collusion, colluders contribute their copies segment by segment (or equivalently, symbol by symbol at the code level) with approximately equal share. Further distortion may be applied on the colluded signal, which we simplify as additive white Gaussian noise. At the detector side, we consider the soft detector employing the matched filter as in (5). With this detector, we skip the symbol detection as in hard detection, and directly identify the colluder by correlating the test signal with every fingerprint sequence. The user whose fingerprint sequence has the highest correlation is declared as colluder. As long as the correlation between the fingerprint sequences is kept low, the performance of the matched-filter decoding approaches that of the maximum-likelihood decoding and provides an upper bound for the ECC-based fingerprinting.

To facilitate further discussions, here we write down the expression of the matched-filter detector again in (8). For each user, we examine a correlation-based statistic T_N as

$$T_N(j) = \frac{(\mathbf{z}-\mathbf{x})^T\mathbf{s}_j}{\sqrt{||\mathbf{s}_j||^2}} \quad j=1,\ldots,N_u \tag{8}$$

which follows a multivariate Gaussian distribution of N_u dimensions. Here, $\mathbf{s}_j$ is the fingerprint sequence for user j, $\mathbf{z}$ is the colluded signal, and $\mathbf{x}$ is the original signal. We define

$$T_1 = \max_{j\in S_C} T_N(j), \quad T_2 = \max_{j\notin S_C} T_N(j) \tag{9}$$

where S_C is the colluder set. For simplicity, we approximate T_1 and T_2 as independent Gaussian variables. By examining the distribution of the correlations between each fingerprint sequence and the test sequence, we can express the mean and the variance of T_1 and T_2 as follows:

$$\begin{aligned} m_{T_1} &\triangleq E[T_1] = \frac{||\mathbf{u}||}{\sqrt{L}}\left(\frac{L}{c} + \frac{5(c-1)(L-D)}{12}\right), \\ \sigma_{T_1}^2 &\triangleq Var[T_1] = \frac{||\mathbf{u}||^2}{L}\sigma_C^2 + \sigma_d^2 \end{aligned} \tag{10}$$

$$\begin{aligned} m_{T_2} &\triangleq E[T_2] = \frac{||\mathbf{u}||}{\sqrt{L}} \times \frac{c(L-D)+1}{2}, \\ \sigma_{T_2}^2 &\triangleq Var[T_2] = \frac{||\mathbf{u}||^2}{L}\sigma_I^2 + \sigma_d^2 \end{aligned} \tag{11}$$

$$\begin{aligned} \text{with} \quad \sigma_I^2 &= \left(\frac{c(L-D)-1}{6}\right)^2, \\ \sigma_C^2 &= \left(\frac{5(c-1)(L-D)}{36}\right)^2 \end{aligned} \tag{12}$$

where σ_d^2 is the variance of the additive noise. Thus, the probability of detection is

$$P_d = Pr(T_1 > T_2) = \int_{-\infty}^{\infty} P(T_1 > t) f_{T_2}(t)dt \tag{13}$$

where f_{T_2} is the pdf of T_2 and

$$P(T_1 > t) = 1 - Q\left(\frac{t - m_{T_1}}{\sigma_{T_1}}\right). \tag{14}$$

2) Averaging Collusion: We employ the matched-filter detector in (8) to analyze the probability of detection under averaging collusion. To get an analytical approximation, we first consider an ideal fingerprinting system whose fingerprint sequences have a constant pairwise correlation, denoted as ρ. Without loss of generality, we assume that the first c users contribute to collusion by performing averaging operations. The vector of detection statistics T_N's defined in (8) and follows an N_u-dimensional Gaussian distribution:

$$\begin{aligned} \mathbf{T} &= [T_N(1), \ldots, T_N(N_u)]^T \\ &\sim N\left([\mathbf{m}_1, \mathbf{m}_2]^T, \sigma_d^2\Sigma\right), \\ \text{with} \quad \mathbf{m}_1 &= ||\mathbf{s}||\left(\frac{1}{c} + \left(1 - \frac{1}{c}\right)\rho\right)\mathbf{1}_c, \\ \mathbf{m}_2 &= ||\mathbf{s}||\rho\mathbf{1}_{N_u-c} \end{aligned} \tag{15}$$

where $\mathbf{1}_k$ is an all-1 vector with dimension k-by-1, Σ is an N_u-by-N_u matrix whose diagonal elements are 1's and offdiagonal elements are ρ's, σ_d^2 is the variance of the noise, $\mathbf{m}_1$ is the mean vector for colluders, and $\mathbf{m}_2$ is the mean vector for innocent users. Given the same colluder number c and fingerprint strength $||\mathbf{s}||$, the mean correlation values with colluders and with innocents are separated more widely for a smaller ρ. This suggests that in absence of any prior knowledge on collusion pattern, a smaller ρ leads to a larger colluder detection probability P_d. Therefore, we prefer fingerprint sequences with a small pairwise correlation ρ in the system design.

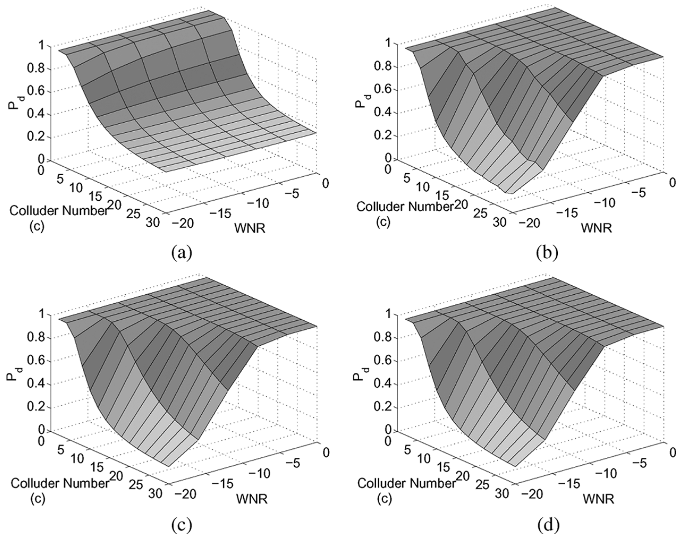
The pairwise correlation of ECC-based fingerprinting can be calculated by examining the code construction. Codes with a larger minimum distance have a smaller upper bound on the correlation and, thus, are more preferable. This is consistent with the principle indicated in (6) to employ codes with a large minimum distance. Under the code construction with a large minimum distance, the largest pairwise correlation ρ_0 between the fingerprinting sequences, which corresponds to the codewords with minimum distance, will be close to 0. We use the above equal pairwise correlation model with $\rho = \rho_0$ to approximate the performance of ECC-based fingerprinting under averaging collusion.

Taking Reed–Solomon code-based fingerprinting as an example, we calculate its pairwise correlation. For an L-tuple q-ary Reed–Solomon code with dimension t, the total number of codewords is $N_u = q^t$ and the minimum distance is $D = L - t + 1$. We use $\mathbf{s}_i$ and $\mathbf{s}_j$ to represent the fingerprint sequences for user i and user j, respectively, and $\mathbf{w}_{ik}$ the orthogonal sequence representing the symbol in user i's codeword at position k with $||\mathbf{w}_{ik}|| = ||\mathbf{w}||$. The normalized correlation between $\mathbf{s}_i$ and $\mathbf{s}_j$ is

$$\begin{aligned} \frac{\langle \mathbf{s}_i, \mathbf{s}_j\rangle}{||\mathbf{s}||^2} &= \frac{\langle[\mathbf{w}_{i1}\mathbf{w}_{i2}\cdots\mathbf{w}_{iL}], [\mathbf{w}_{j1}\mathbf{w}_{j2}\cdots\mathbf{w}_{jL}]\rangle}{L||\mathbf{w}||^2} \\ &\le \frac{L-D}{L} = \frac{t-1}{L} = \rho_0. \end{aligned} \tag{16}$$

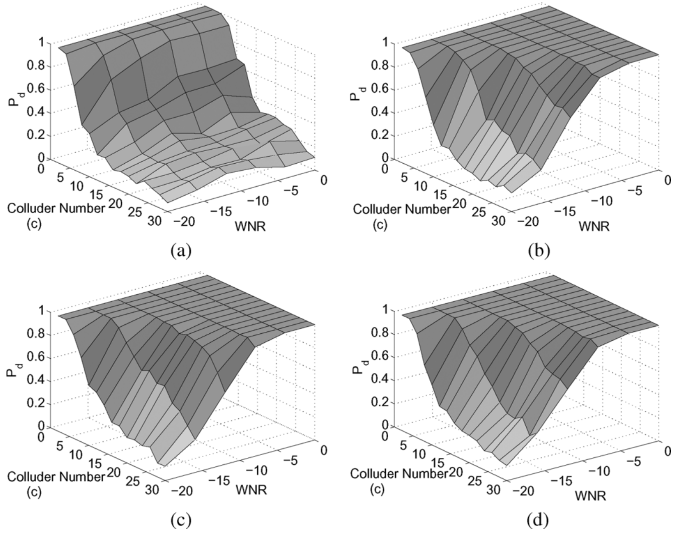
We can choose t and L such that the correlation ρ_0 is close to 0. By doing so, the ECC-based fingerprinting and the orthogonal fingerprinting should have comparable resistance against averaging collusion.

3) Numerical Results: In order to illustrate the collusion resistance derived from the above analysis, we consider an example system with the parameters chosen as follows. For a system holding N_u users, the results in (7) and (16) show that a larger L and a smaller t are preferred in order to get better collusion resistance under interleaving and averaging collusion. Because t can only take integer values, we take $t = 2$



(a) (b) (c) (d)

Fig. 2. Analytical approximation of ECC-based fingerprinting under (a) interleaving collusion, (b) averaging collusion, and orthogonal fingerprinting under (c) interleaving collusion, and (d) averaging collusion.



(a) (b) (c) (d)

Fig. 3. Simulation results of ECC-based fingerprinting under (a) interleaving collusion, (b) averaging collusion, and orthogonal fingerprinting under (c) interleaving collusion, and (d) averaging collusion.

to obtain a nontrivial Reed–Solomon code construction. This also determines q since $q^t = N_u$. On the other hand, larger L results in a smaller segment size for a given host signal, which may lead to a higher error probability in symbol detection. Typically a segment size of 1000 can provide reliable symbol detection. With an additional condition that $L \leq q$, we choose L to be a number smaller than but close to q. In our example, considering a total of $N_u = 1024$ users and a host signal with $N = 3 \times 10^4$ embeddable components, we choose $L = 30$ and use a Reed–Solomon code with parameters of $q = 32$ and $D = 29$. According to (6), the code level alone can only assure resisting up to five users' interleaving collusion; on the other hand, the correlation between fingerprint sequences is only 0.03 according to (16), which suggests it should have similar performance to orthogonal fingerprinting under averaging collusion.

We show the analytical approximation of P_d for the ECC-based fingerprinting under interleaving and averaging collusion with the above settings in Fig. 2(a) and (b), respectively. The watermark-to-noise-ratio (WNR) ranges from 0 to -20 dB, which includes the scenarios from severe distortion to mild distortion. The theoretical results for orthogonal fingerprinting from [2] are shown in Fig. 2(c) and (d) for interleaving collusion and averaging collusion, respectively. Comparing Fig. 2(b) and (d), we see that under averaging collusion, the orthogonal fingerprinting and the ECC-based fingerprinting constructed above have similar colluder identification performance. They both can resist at least a few dozen's colluders' averaging attack under high WNR and about half a dozen's under very low WNR. This is consistent with the above analysis of the collusion resistance against averaging collusion. Thus, from the colluders' point of view, the averaging collusion for an ECC-based fingerprinting system is not a very effective strategy. However, under interleaving collusion, we observe from Fig. 2(a) and (c) a huge gap on the collusion resistance between the two systems. For orthogonal fingerprinting, the probability of colluder detection under interleaving collusion is the same as that under averaging collusion owing to the orthogonal spreading; at $\mathrm{WNR} = 0$ dB, the P_d remains close to 1 when c is around a few dozen. On the other hand, the detection probability of the ECC-based fingerprinting drops sharply when more than seven colluders come to create an interleaved copy, even when WNR is high. Thus, from a colluders' point of view, interleaving collusion is an effective strategy to circumvent the protection.

To validate the analysis, we apply both systems to a host signal that is modeled as an i.i.d. Gaussian sequence with length $N = 3 \times 10^4$. This simple assumption on the host signal suits the fingerprinting applications well since the host signal is often known to the detector, and its effect will be mostly removed by subtracting it from the colluded signal. As such, the distribution of the host signal does not have a major effect on the detection performance. The detector in (8) is employed for both fingerprinting systems. We measure the probability of correctly catching a colluder (P_d) for different values of colluder number c. The results of 200 iterations are shown in Fig. 3. Notice that the analytical approximation of ECC-based fingerprinting under interleaving collusion [Fig. 2(a)] is higher than the measured value of P_d for large c. This is because the analysis in (10)–(12) considers the maximum number of matched symbols between the colluded codeword and an innocent codeword as $c(L-D)$. Using such an assumption to estimate P_d becomes less accurate for large c. However, the analytical approximation captures the trend and provides an upper bound for the P_d of ECC-based fingerprinting under interleaving collusion. All other analytical results match well with the simulation results. In summary, the simulation results verify the analytical approximation derived for interleaving collusion and averaging collusion and validate the conclusions drawn from the analytical results.

When designing a fingerprinting system, a better tradeoff between the collusion resistance and other performance measures, such as detection computational complexity, is desired. Although orthogonal fingerprinting performs well in collusion resistance, its detection computational complexity and distribution cost are expensive as we have seen in Sections III-A and III-B. The significant computational and distribution advantages of ECC-based fingerprinting motivate us to find avenues to improve its collusion resistance, especially to reduce the performance gap between the ECC-based fingerprinting and orthogonal fingerprinting while preserving its efficient detection and distribution. In the following sections, we identify two directions for improving collusion resistance and propose two new techniques that jointly consider coding and embedding of fingerprint, namely, permuted subsegment embedding and GRACE fingerprinting.

IV. Permuted Subsegment Embedding Technique

A. Proposed Embedding Method

The drastic difference in the collusion resistance against averaging and interleaving collusions of ECC-based fingerprinting inspires us to look for an improved fingerprinting method, for which the interleaving collusion would have a similar effect to averaging collusion. Careful examination on the two types of collusions shows that the difference in the resistance against them comes from the amount of role given to the embedding layer to play. The segment-wise interleaving collusion is equivalent to the symbol-wise interleaving collusion on the code level since each colluded segment comes from just one user. The collusion resilience primarily relies on what is provided by the code layer and almost bypasses the embedding layer. Because of the limited alphabet size, the chance for the colluders to interleave their symbols and create a colluded fingerprint close to the fingerprint of an innocent user is so high that it would require a large minimum distance in the code design, if to handle this on the code level alone. This means that either codes representing a given number of users can resist only a small number of colluders, or codes can represent only a small total number of users. On the other hand, for the averaging collusion, every colluder contributes his or her share in every segment. Through a correlation detector, the collection of such a contribution over the entire test signal leads to high expected correlation values when correlating with the fingerprints from the true colluders, and to low expected correlation values when with the fingerprints from innocent users. In other words, the embedding layer contributes to defending against the collusion. This suggests that more closely considering the relation between fingerprint encoding, embedding, and detection is helpful to improve the collusion resistance against interleaving collusion.

The basic idea of our improved algorithm is to prevent the colluders from using the whole segment that carries one symbol as an interleaving unit and to exploit the code-level limitation. We accomplish this by making each colluded segment contain multiple colluders' contribution. Our solution builds upon the existing code construction and performs two important additional steps that we collectively refer to as permuted subsegment embedding [24]. As shown in Fig. 4, consider as before a fingerprint signal generated by concatenating the appropriate sequences corresponding to the symbols in a user's codeword. We first partition each segment of the fingerprint signal into β subsegments, giving a total of βL subsegments. We then randomly permute these subsegments according to a secret key to obtain the final fingerprint signal to represent the user. In detection, the extracted fingerprint sequence is first inversely permuted and then the correlator (8) is applied to identify the colluder.

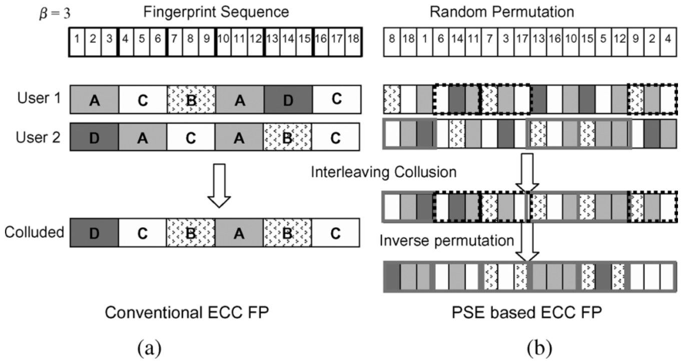


Fig. 4. Illustration of the permutated subsegment embedding for ECC-based fingerprinting. (a) The conventional ECC-based fingerprinting. (b) The proposed scheme.

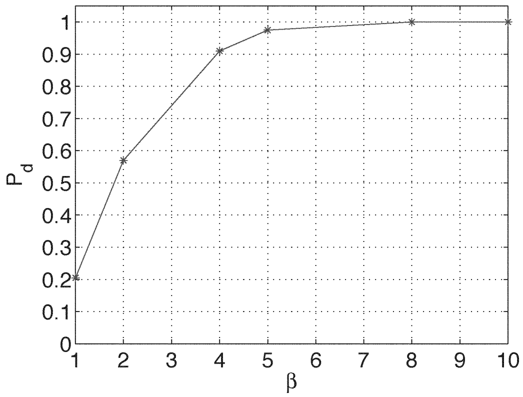
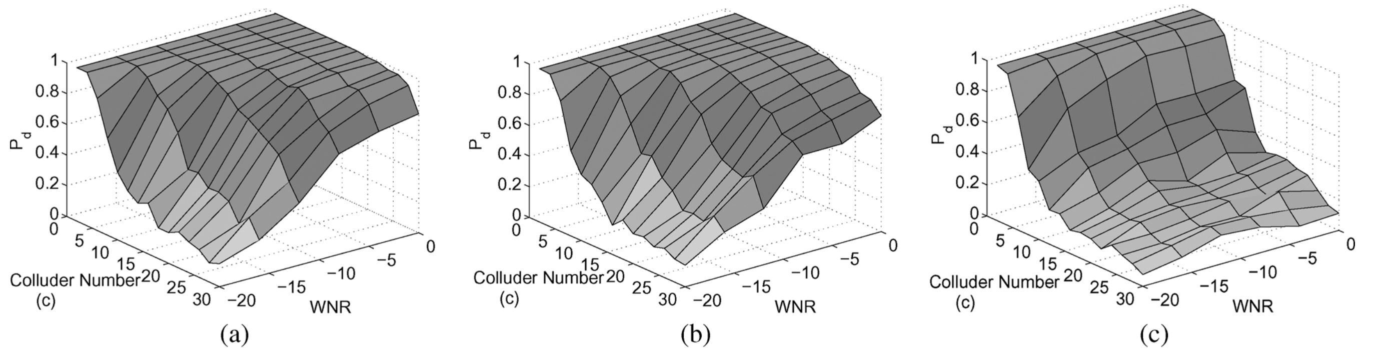


Fig. 5. Probability of catching one colluder P_d versus β for $c = 25$ and $\mathrm{WNR} = 0$ dB of the proposed scheme.

With subsegment partitioning and permutation, each colluded segment after interleaving collusion most likely contains subsegments from multiple users. To correlation-based detectors (including both hard and soft detection on the symbol level), this would have a similar effect to what averaging collusion brings. Since averaging collusion is far less effective from the colluders' point of view, the permuted subsegment embedding can greatly improve the collusion resistance of ECC-based fingerprinting under interleaving collusion. Even if the colluders know the actual size of a segment or a subsegment, the permutation unknown to them prevents them from creating a colluded signal with the equivalent effect of symbol interleaving in the code domain.

The detection statistic T_N for the improved system under interleaving collusion can be approximated by an N_u-dimension Gaussian distribution

$$
\begin{aligned}
& T_N \sim N\left([\mathbf{m}_1, \mathbf{m}_2], \sigma_d^2 \Sigma\right) \\
\text{with}\quad & \mathbf{m}_1 = \left(\left\lfloor \frac{\beta L}{c} \right\rfloor + \left\lfloor \frac{L-D}{L}\left(\beta L - \left\lfloor \frac{\beta L}{c} \right\rfloor\right)\right\rfloor\right) \frac{\|\mathbf{s}\|}{\beta L} \\
& \mathbf{m}_2 = \left\lfloor \frac{L-D}{L} L\beta \right\rfloor \frac{\|\mathbf{s}\|}{\beta L} = \frac{L-D}{L}\|\mathbf{s}\| \qquad (17)
\end{aligned}
$$



(a) (b) (c)

Fig. 6. Collusion resistance of the improved ECC-based fingerprinting with permuted subsegment embedding technique under (a) segment-wise and (b) subsegment-wise interleaving collusion; (c) collusion resistance of the conventional ECC-based fingerprinting under interleaving collusion.

where Σ is the same as that in (15) under averaging collusion with $\rho = (L-D)/L$. We can see that the parameter β controls the "approximation" level of the effect of interleaving collusion to that of averaging collusion. Larger β provides a finer granularity in subsegment division and permutation. Thus, each segment may contain subsegments from more colluders, leading to better approximation and better collusion resistance. We verify this relation by building an improved ECC-based fingerprinting system with different β values upon the experiment setup in Section III-C. Fig. 5 shows the results when a total of $c = 25$ colluders perform segment-wise interleaving with WNR$= 0$ dB. We can see that higher β indeed gives higher detection probability P_d. On the other hand, a larger β may incur higher computational complexity in permutation. Thus, a tradeoff should be made according to the requirements of a specific application. Notice that for the particular system we examined in Fig. 5, the improvement on the detection probability saturates when $\beta > 5$. Therefore, we choose $\beta = 5$ for this system in later experiments to obtain a good tradeoff between the permutation computational complexity and the detection performance improvement.

B. Experimental Results

We evaluate the performance of the improved system with $\beta = 5$ under various WNRs, and show the results in Fig. 6(a) for segment-wise interleaving collusion. For comparison, we show the performance of the conventional ECC-based fingerprinting under segment-wise interleaving collusion in Fig. 6(c). We can see that the detection probability of the proposed system is substantially improved over the conventional ECC-based fingerprinting system under the same interleaving collusion. Under around two dozens users' collusion, the probability of detection P_d increases up to four times that of the conventional ECC-based fingerprinting at high and moderate WNRs. In the meantime, the gap between the performance of the proposed system in Fig. 6(a) and that of the orthogonal fingerprinting in Fig. 3(c) is very small.

Next, Fig. 6(b) shows the results for interleaving collusion using subsegment as a unit. We observe from Fig. 6(a) and (b) that when many users come together to perform interleaving collusion (i.e., for large c), the performance of the proposed system is a little worse when the interleaving is done using a subsegment as a unit than that when using a segment as a unit. This is because the probability that one segment contains only one colluder's trace after subsegment interleaving and inverse permutation is a little higher than that after segment interleaving. As we pointed out earlier, one segment containing more colluders' information after the collusion leads to a higher performance in colluder detection. As such, the collusion resistance against subsegment interleaving is slightly worse than that against segment interleaving. Overall, the proposed system has similar performance under two types of interleaving collusion and gives a high detection probability for up to two dozen colluders at moderate-to-high WNR. Since the permuted subsegment embedding does not affect the performance of the system under averaging collusion, the P_d under averaging collusion remains unchanged. We can see that the proposed system based on the joint consideration of the fingerprint coding and embedding has effectively improved the collusion resistance.

C. Discussions

1) Role of Permutation: Random permutation is a useful technique that has found quite a few applications in data embedding. It was used in image watermarking to equalize the uneven embedding capacity [25], and was applied to a simple staircase construction of binary fingerprint code to prevent framing innocent users [7]. In our proposed work, we employ random permutation to make each segment after interleaving collusion contain multiple colluders' information, thus mimicking the effect of averaging collusion, and improving the collusion resistance against interleaving collusion.

2) Computational Complexity of Fingerprint Detection and Efficient Distribution: The detection of the improved ECC-based fingerprinting using permuted subsegment embedding consists of three steps: inverse permutation, demodulation by correlation, and decoding to certain colluder. The computational complexity of the inverse permutation is $O(\beta L)$. As we have analyzed in Section III-A, the other two steps need at most $O(qN)$ computations. Thus, the improved ECC fingerprinting has the complexity of $O(\beta L) + O(qN)$. Since the largest possible value of βL is the total number of the embeddable components N, the demodulation step still dominates the overall complexity. Therefore, the overall computational complexity remains at $O(qN)$.

Notice that in the improved ECC-based fingerprinting, for each subsegment, there are only q different versions. The efficient distribution of the fingerprinted signal discussed earlier for

TABLE I
PERFORMANCE COMPARISON OF FINGERPRINTING SYSTEMS

	Orthogonal Fingerprinting	Improved ECC Fingerprinting	ECC Fingerprinting
Collusion Resistance to interleaving collusion (Number of colluders)	One order of magnitude more than the number of ECC FP	Between ECC and Orth FP. Approach to averaging collusion for large β.	On the order of $\sqrt{q}$
Collusion Resistance to averaging collusion (Number of colluders)	Same as interleaving collusion	Similar to Orth FP for small ρ	Similar to Orth FP for small ρ
Detection Computational Complexity	$O(N_u N)$	$O(\sqrt[t]{N_u} N)$	$O(\sqrt[t]{N_u} N)$
Distribution Efficiency γ	1	q^{1-t}	q^{1-t}

ECC-based fingerprinting is still applicable here except that the multicast becomes subsegment based instead of segment based. While the bandwidth efficiency (in terms of the cost ratio γ defined earlier) remains unchanged, the multicast groups have to be updated when transmitting each subsegment. The more subsegments (or larger β) we have, the more frequently we have to switch the multicast grouping. This overhead should be taken into account when choosing β.

3) Comparison Criteria: The results in Fig. 6 show that the proposed permuted subsegment embedding provides significant collusion resistance improvement for ECC-based fingerprinting with only a small increase of computation and distribution cost. Moreover, different user-capacity requirements can be accommodated by preserving the alphabet size and adjusting the dimension of the ECC. For Reed–Solomon code, this can be done by adjusting the parameter t. We summarize in Table I the collusion resistance, detection, and distribution efficiency for three fingerprinting systems, namely, ECC-based fingerprinting ("ECC FP" in short), improved ECC-based fingerprinting with permuted subsegment embedding, and orthogonal fingerprinting ("Orth FP" in short). Overall, the improved ECC-based fingerprinting provides a better tradeoff among these three criteria over the conventional schemes, and offers flexibility to accommodate different application requirements.

It is worth noting that the comparison that we have seen is the resistance against averaging collusion and interleaving collusion at the same WNR. Under such settings, we have found that interleaving collusion is a more effective attack than averaging collusion. We thus focus on improving the system's resistance against interleaving collusion, and propose the permuted subsegment embedding technique to bring similar performance against both types of collusions. Another possible comparison setting is to keep the same mean square error (MSE) of the colluded signal with respect to the original signal for both types of collusions. Notice that for fingerprint sequences with small correlation, averaging operation brings the colluded signal (before additive noise and other further distortions) close to the original signal. As such, for the same level of overall MSE distortion, averaging collusion allows stronger noise to be added than interleaving collusion does. In this sense, averaging collusion may become more effective than interleaving collusion after permuted subsegment embedding, especially when the number of colluders is large. The detailed colluder tracing results under this alternative setting can be obtained by mapping the WNR in Fig. 6 to the corresponding MSE distortion. One aspect to be taken into account is the limitation of MSE in reflecting the true perceptual effect. Averaging collusion plus additive noise does not necessarily render the same level of imperceptibility as interleaving collusion, especially when the noise is random and does not match the multimedia content. We will explore this problem further in our future work.

V. GRACE: GROUP-BASED JOINT CODING AND EMBEDDING FINGERPRINTING

Our second improvement technique is rooted from the observation that a user is often not equally likely to collude with other users in practice. For example, users in the same geographic area or having similar social or cultural background may be more likely to collude. Taking advantage of this prior knowledge, Wang *et al.* proposed group-oriented fingerprinting to enhance the collusion resistance of noncoded orthogonal fingerprinting [3]. In their work, users are put into groups according to the group collusion behavior, and each user's fingerprint consists of two parts of information identifying each individual user as well as the group he or she is in. The group information is used in the detection to narrow down the suspicious user set. Such prior knowledge of the collusion pattern has not been exploited in the coded fingerprinting, where new issues arise, such as how to group users and how to construct and embed the group information and user information.

In the meantime, the results in Section III-C suggest that the performance of the conventional ECC-based fingerprinting is mainly restricted by the code structure especially for high WNR where the symbol detection from the embedding layer has high accuracy. For example, we see from Fig. 3(a) that as WNR increases from -20 to 0 dB, the detection probability of the ECC-based fingerprinting only increases 0.1–0.15 compared with the huge increase of 0.7–0.8 in orthogonal fingerprinting. Based on this observation, it is possible to use part of the fingerprint energy to embed group information to facilitate the colluder detection, while keeping the symbol detection accuracy high enough. We thus propose the GRACE fingerprinting system [26]. In the GRACE fingerprinting, we construct the fingerprint sequence by superposing the sequences for the group information and the user codeword. This combined fingerprint is spread over the host signal during embeddding. As we shall see, this joint coding and embedding significantly improves the collusion resistance of the ECC-based fingerprinting.

A. Fingerprint Construction and Embedding

We partition the codewords in ECC based fingerprinting into groups to capture the collusion pattern, and assign symbols to each group to represent the group information. We call these group symbols "group subcode", and refer to the symbols for distinguishing individual users as "user subcode". Thus each user's fingerprint consists of two parts, namely, user subcode and group subcode.

Algorithm 1: Group construction in GRACE fingerprinting

```
1) Set the group index $i = 1$, initialize the set of codewords
   for group $i$ to be empty $G(i) = \varnothing$;
2) Pick any codeword $c \in C$ to be the first element for
   group $i$, move it from $C$ to group $i$: $G(i) = \{G(i), c\}$,
   $C \leftarrow C - \{c\}$;
3) Examine every codeword in $C$: If $c \in C$ is orthogonal
   to all the existing codewords in $G(i)$, move $c$ from $C$ to
   $G(i)$;
4) If $C \neq \varnothing$, continue to build the next group. Set $i \leftarrow i + 1$,
   initialize $G(i) = \varnothing$, and go to step 2.
```

1) Subcode Construction: To construct the user subcode, we start with a c-TA code based on error correcting code construction over an alphabet of size q as discussed earlier in Section II. The code length is L, and the minimum distance is D and, typically, less than L. We then rearrange the codebook into groups so that within each group, the codewords are orthogonal to each other (i.e., users within the group have distinct values at each symbol position). Thus, the code distance within a group equals the codeword length L. We assign one codeword to each user as his or her user subcode. This process is described in more detail in Algorithm 1. Other construction of orthogonal subcodes is also possible, for example, through a systematic coding technique known as mutually orthogonal Latin squares (MOLT) [27].

Next, we construct the group subcodes. To make group information as separate as possible and, thus, facilitate accurate identification of guilty groups, we design the group subcodes to be orthogonal to each other. A simple way to construct the group subcode is to use one distinct symbol to represent one group; thus, we need a total of g symbols for g groups. For each group, we construct repetition code with length L by repeating the symbol L times as the group subcode.

2) Fingerprint Embedding: In the proposed GRACE fingerprinting scheme, we embed both group subcode and user subcode by mapping them to spreading sequences and then adding the superposition of the two corresponding spreading sequences to the host signal.

The group information of the GRACE fingerprinting is orthogonal to the spreading sequence conveying the user subcode, yet their supports overlap in the signal sample domain [16]. More specifically, we use the sequences $\{\mathbf{u}_j, j = 1, \ldots, q\}$ to represent q symbol values in the alphabet of user subcode, where $\mathbf{u}_j$'s are orthogonal to each other and have identical energy $||\mathbf{u}||^2$. The g sequences $\{\mathbf{a}_i, i = 1, \ldots, g\}$ represent g groups. They are orthogonal to each other and to $\{\mathbf{u}_j\}$, and have the same energy as $\mathbf{u}_j$'s (i.e., $||\mathbf{a}||^2 = ||\mathbf{u}||^2$). We then construct the fingerprint sequence for the kth segment of user j who belongs to group i as

$$\mathbf{s}_{ijk} = \sqrt{1-\rho}\,\mathbf{u}_{\mathrm{sym}(j,k)} + \sqrt{\rho}\,\mathbf{a}_i \tag{18}$$

where the function $\mathrm{sym}(j,k)$ is used to retrieve the symbol for the kth segment from the jth user's subcode, and ρ is used to adjust the relative energy between the group subcode and user subcode. This fingerprint signal is finally added to the kth segment of the host signal. A larger ρ puts more energy on group information and, thus, provides a more accurate detection of group information. However, a larger ρ also reduces the detection accuracy of user subcode and makes it harder to narrow down to the true colluder. Therefore, there is a tradeoff between group detection and user detection when choosing ρ. Since in our scheme, we have L segments to collect the energy for group detection, and usually collusion occurs among a small number of groups, we can choose a small ρ to satisfy the detection performance requirement on both user information and group information.

We can see that a key design issue in the GRACE fingerprinting is on how to represent and embed the group information versus the user information. Our approach is to superpose the spreading sequences of group subcode and user subcode for embedding. Alternatively, the group information may be embedded by appending the spreading sequence of group subcode to that of user subcode. To demonstrate the performance gain of the GRACE fingerprinting brought by the joint consideration of coding and embedding, we shall present this appending scheme as well and refer to it as the group ECC fingerprinting by appending. In this alternative fingerprinting scheme, the equivalent codeword for each user is the concatenation of the user subcode with length L and the group subcode with length L_g, where L_g is not necessarily equal to L and is used to adjust the relative energy between the group subcode and the user subcode. The total codeword length is $L + L_g$. To embed this codeword, the host signal is partitioned into $L + L_g$ segments. The corresponding spreading sequence is added into each segment according to the codeword symbols. For a given host signal where the total number of embeddable signal samples N is fixed, the longer the group subcode is, the smaller the length each segment $N'_s = N/(L + L_g)$ is.

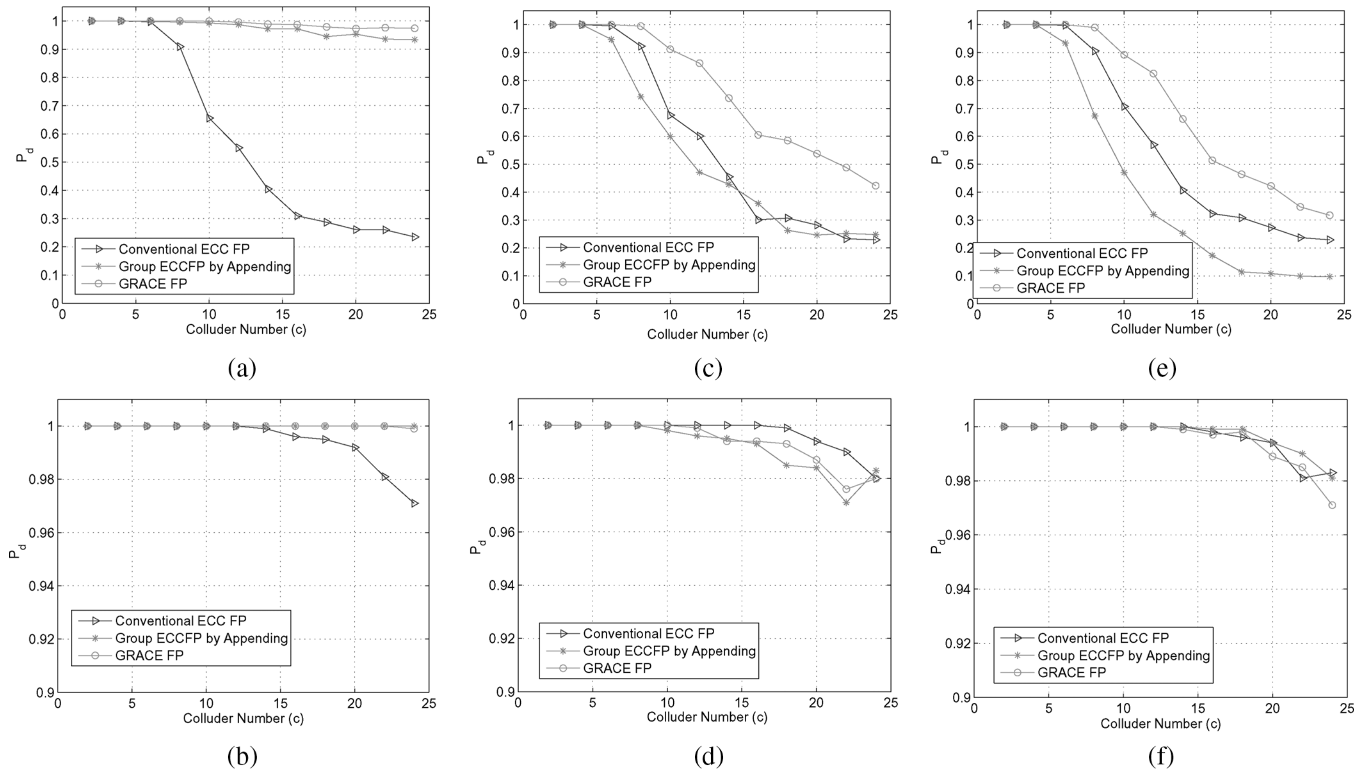
B. Fingerprint Detection

At the detector side, the embedded group information can be used to facilitate the detection by a two-level detection scheme. First, we examine through a correlation detector the group information in the colluded signal to identify the groups from which the colluders come. We then focus our attention on these identified suspicious groups and apply matched-filter detection for ECC-based fingerprinting as discussed in Section II on the user subcode to narrow down to the true colluders.

More specifically, we extract group information from the colluded signal $\mathbf{z}$ using a nonblind correlation detector. The detection statistic with respect to group i is

$$T_G(i) = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{b}_i}{||\mathbf{b}||}, \quad i = 1, 2, \ldots, g \tag{19}$$

where $\mathbf{x}$ is the host signal, and $\mathbf{b}_i$ is the concatenation of the spreading sequences representing group i's information from each segment. In the above settings, $\mathbf{b}_i^T = [\mathbf{a}_i^T, \ldots, \mathbf{a}_i^T]$ since we embed $\mathbf{a}_i$ in each segment of group i. The kth group is considered guilty for the test signal if $T_G(k) > h$, where h is the threshold. The union of the detected groups forms a suspicious group set. To narrow down to the true colluders inside the suspicious groups, we employ the soft detector in (5) to correlate



(a) (c) (e)

(b) (d) (f)

Fig. 7. Performance comparison of the proposed GRACE fingerprinting, group ECC fingerprinting by appending and the conventional ECC fingerprinting schemes in terms of probability of detection P_d versus the colluder number c at $\mathrm{WNR} = 0\ \mathrm{dB}$. (a) Two-group interleaving collusion. (b) Two-group averaging collusion. (c) Random-group interleaving collusion. (d) Random-group averaging collusion. (e) Distinct-group interleaving collusion. (f) Distinct-group averaging collusion.

the test signal with each user's fingerprint sequence and identify the one with the highest correlation as the colluder.

The detection for the group ECC fingerprinting by appending is a two-stage process similar to GRACE fingerprinting. We first extract the group information from the segments corresponding to group subcode through a nonblind correlation detector. The decoding to a specific colluder is then conducted on the segments for user subcode within the extracted suspicious groups.

C. Experimental Results

In this section, we demonstrate the effectiveness of the proposed GRACE fingerprinting through experiments. To build the user subcode, we employ a Reed–Solomon code with $q = 32$, $L = 30$, $N_u = 1024$, $D = 29$, and rearrange it into 32 groups using the algorithm described in Section V-A. Inside each group, there are 32 codewords mutually orthogonal to each other. We choose $\rho = 1/7$ in (18) to generate the fingerprint signal from the user subcode and the group subcode in GRACE. For fair comparison, we choose $L_g = 5$ for the group ECC fingerprinting by appending it in order to provide the same relative energy between user subcode and group subcode as that of GRACE. We use the repetition code described in Section V-A as the group subcode, and construct i.i.d. Gaussian signals with 3×10^4 signal samples to emulate the host signal.

Interleaving collusion and averaging collusion are applied to all three systems, namely the ECC-based fingerprinting, the GRACE fingerprinting, and the group ECC fingerprinting by appending. We examine the probability of successfully detecting one colluder (P_d) at $\mathrm{WNR} = 0\ \mathrm{dB}$ in the following three scenarios:

1) Collusion Within a Small Number of Groups: In this case, our grouping correctly reflects the collusion pattern that all of the colluders come from a small number of groups. In our simulation, all colluders are from two out of 32 groups, and they are randomly distributed between these two groups. The results of P_d under interleaving collusion and averaging collusion are shown in Fig. 7(a) and (b), respectively. Under interleaving collusion, we can see that for the same number of colluders, the P_d's for the proposed GRACE and the group ECC fingerprinting by appending are similar, and they have up to 0.7 improvement over that of the conventional ECC-based fingerprinting. From another point of view, if we require the P_d of the system to be no less than a given value, say 0.98, the number of colluders that the system can resist can be improved from six colluders (for conventional ECC-based fingerprinting) to 18 colluders (for the proposed GRACE fingerprinting). Under the averaging collusion, all systems have P_d close to 1 for the examined c values, but we still can see 0.02 improvement on P_d brought by GRACE fingerprinting over the conventional ECC fingerprinting.

2) Colluders Randomly Distribute Across All Groups: In this case, the grouping does not capture the collusion pattern. The colluders randomly distribute across all groups. The results under interleaving and averaging collusion are shown in Fig. 7(c) and (d), respectively. Under interleaving collusion, the proposed GRACE fingerprinting has up to 0.3 improvement on P_d over the conventional ECC fingerprinting, while the alternative technique of group ECC fingerprinting by appending performs

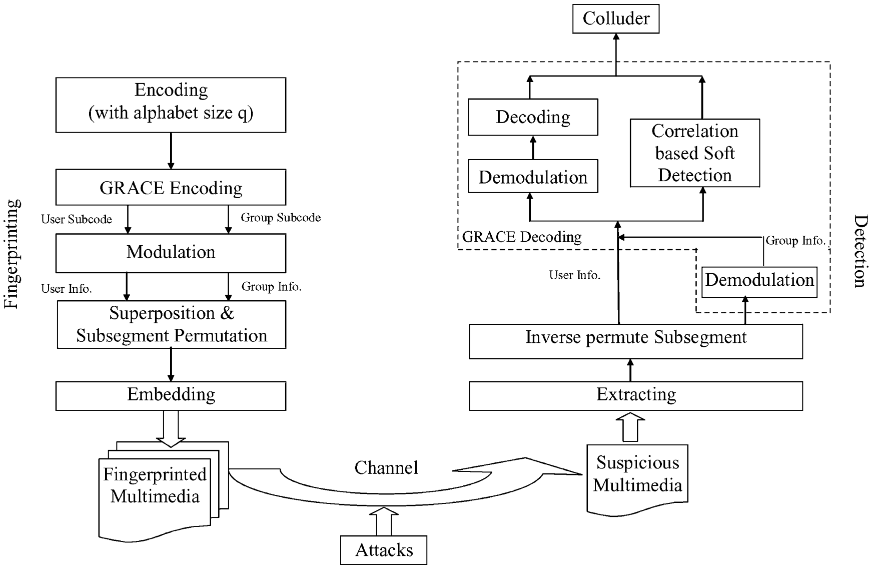


Fig. 8. Proposed framework of coded multimedia fingerprinting combining GRACE with permuted subsegment embedding.

a little worse than the conventional ECC fingerprinting. Under averaging collusion, the proposed GRACE fingerprinting has comparable performance with the ECC-based fingerprinting.

3) Colluders Come From Distinct Groups: In this case, the grouping knowledge is extremely inaccurate. All of the colluders come from distinct groups (i.e., the number of groups equals the number of colluders c). The results under interleaving and averaging collusion are shown in Fig. 7(e) and (f), respectively. Under interleaving collusion, the proposed GRACE fingerprinting still has up to 0.2 improvement on P_d over the conventional ECC fingerprinting. The group ECC fingerprinting by appending performs worse than the conventional ECC fingerprinting with about 0.15 less on P_d. Under averaging collusion, the proposed GRACE fingerprinting has comparable performance with the conventional ECC fingerprinting.

The above results can be explained as follows. When collusion occurs within a small number of groups, the group information is well preserved so that the group detection for both GRACE fingerprinting and the group ECC fingerprinting by appending has high accuracy. As the user subcodes within a small number of groups can be well distinguished due to higher minimum distance than that of the whole codebook, the colluder detection is more accurate than that of the nongroup case. When colluders come from multiple groups or even distinct groups and apply interleaving collusion, the energy of the group subcode for GRACE fingerprinting is reduced after collusion but does not completely diminish because of the spreading of group information over the entire host signal. Therefore, we still have some improvement in detection, although it is not as much as the first case.

For group ECC fingerprinting by appending, when the number of groups gets larger, especially larger than L_g, it is likely that only part of the colluders contribute the group subcode after segment-by-segment interleaving collusion. The detector loses the information of some guilty groups, which leads to no performance improvement over the ECC-based fingerprinting. In contrast, the group information from all colluders can be retained for the two group-based schemes when colluders perform averaging operations, leading to the similar performance by the two schemes. When multiple groups participate in the collusion as in the scenarios 2) and 3), the energy of the group information is reduced by averaging. As such, the group detection has low accuracy, resulting in the diminishing performance gain over ECC-based fingerprinting.

The comparison between the GRACE fingerprinting and the group ECC fingerprinting by appending demonstrates the performance improvement that can be achieved by the joint consideration of coding and embedding. Without the joint consideration, the group ECC fingerprinting by appending is equivalent to the code-level grouping. Separating group information and user information makes it vulnerable to multiple groups' interleaving collusion. In contrast, the proposed GRACE fingerprinting leverages the embedding layer to spread the group information over multiple segments. This helps retain the group information after collusion attacks and, thus, helps identify the true colluders. In addition to $\mathrm{WNR} = 0\ \mathrm{dB}$ presented in Fig. 7, we also examined the cases of low WNRs, and the comparative results are similar to the high WNR case. Overall, the joint coding and embedding as well as the grouping in the proposed GRACE system have brought consistent performance improvement over the existing ECC-based fingerprinting under various scenarios.

D. Combining GRACE With Permuted Subsegment Embedding

Earlier in Section IV, we proposed a new permuted subsegment embedding technique for ECC-based fingerprinting, which improves the collusion resistance while retaining the efficiency in detection and distribution. We can combine the permuted subsegment embedding and the GRACE fingerprinting to arrive at a complete design of the coded fingerprinting system as shown in Fig. 8. We envision that the combined design can provide further improvement on collusion resistance and we will verify it through experiments.

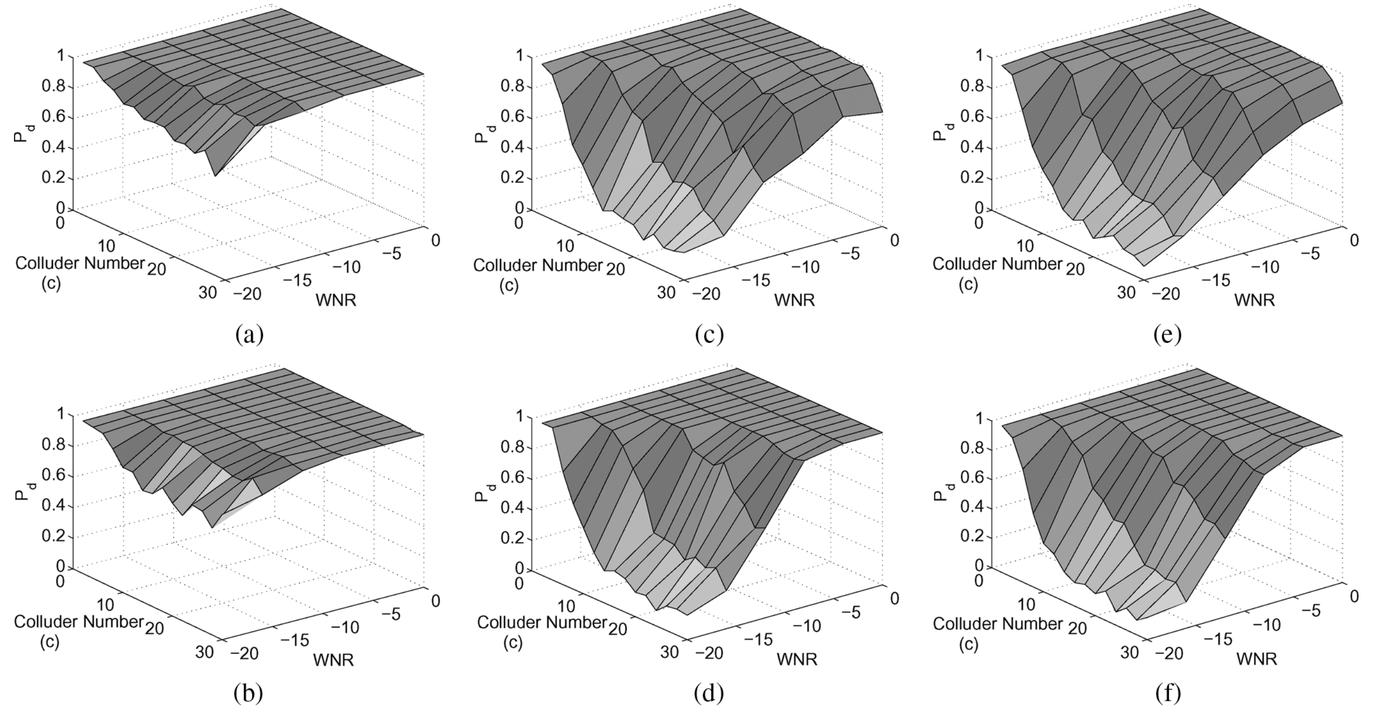


Fig. 9. Performance of the proposed GRACE fingerprinting with permuted subsegment embedding technique: probability of detection P_d versus the colluder number c and WNR. (a) Two-group interleaving collusion. (b) Two-group averaging collusion. (c) Random-group interleaving collusion. (d) Random-group averaging collusion. (e) Distinct-group interleaving collusion. (f) Distinct-group averaging collusion.

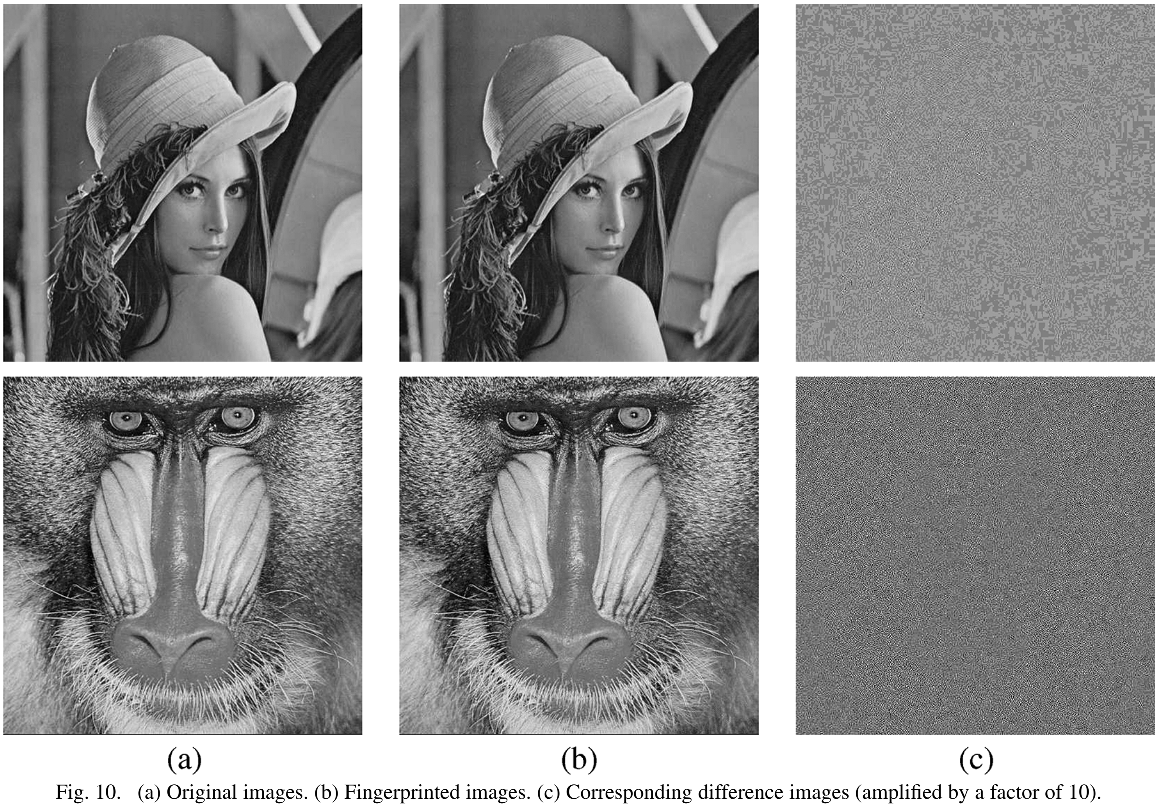


Fig. 10. (a) Original images. (b) Fingerprinted images. (c) Corresponding difference images (amplified by a factor of 10).

In the combined design, the fingerprint sequence of the group subcode is superposed with that of the user subcode as before. We then employ the permuted subsegment embedding to embed the superposed fingerprint sequence to the host signal. A two-level detector is employed after the inverse permutation at the detector side, namely, the extraction of the group information followed by the soft detection of the colluder using (5) within the extracted groups. We demonstrate the performance of the combined fingerprinting system through simulations on the same system as we have examined in the previous sections.

As we have expected, the combination of the proposed two approaches achieves better results than each individual approach. In the cases with inaccurate grouping information [Fig. 9(c)–(f)], the permuted subsegment embedding further

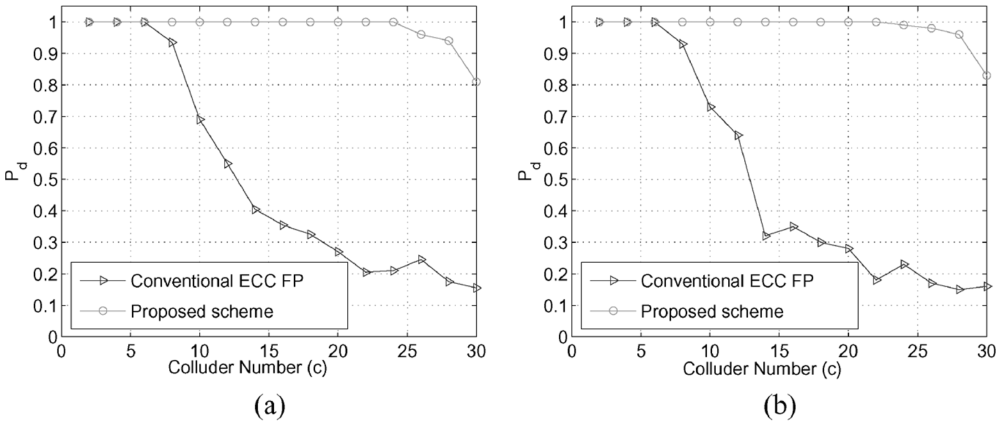


Fig. 11. Experimental results on real images of (a) Lena and (b) Baboon under interleaving collusion.

improves the detection probability P_d of the fingerprinting system by 0.4–0.5 under interleaving collusion at high WNR. The combined design can resist up to 25 users' collusion with high probability of detection, which is more than three times as many as that of the conventional ECC fingerprinting. When the grouping is accurate [Fig. 9(a) and (b)], the grouping strategy boosts the detection probability P_d to nearly 1 for a wide range of WNR and c.

In order to further demonstrate the effectiveness of the proposed joint-coding-and-embedding techniques, we apply the combination of the two newly proposed approaches to natural images and compare its collusion resistance performance with that of the conventional ECC fingerprinting. We use the transform-domain SS scheme for fingerprint embedding, where the original image is divided into 8 × 8 blocks and the fingerprint signal is added into the block DCT coefficients after perceptual weighting. The fingerprint basis is generated according to i.i.d. Gaussian distribution $N(0, 1)$. In this experiment, we perform nonblind detection where the original host signal is available and subtracted from a test signal.

We select 512 × 512 Lena and Baboon as original images to demonstrate the performance of the proposed fingerprinting system on images with different natures. We apply two schemes on both images and examine their performance under collusion attacks: one is the conventional, nongrouped ECC-based fingerprinting scheme, and the other is our proposed GRACE fingerprinting scheme with permuted subsegment embedding. We employ the same coding setup as in Section III for these two images (i.e., Reed–Solomon code of length 30, dimension 2, and minimum distance 29). The effective segment size is 2189 for Lena and 4740 for Baboon. The fingerprinted images have an average PSNR of 41.6 dB for Lena and 33.2 dB for Baboon. Fig. 10 shows the original and fingerprinted images along with the corresponding pixel-wise difference between them.

We examine the scenario of interleaving collusion by randomly distributed colluders across all groups with $\mathrm{WNR} = 0$ dB. The results of 100 iterations on the two images are shown in Fig. 11, where the number of colluders the system can resist is increased from 6 for conventional ECC fingerprinting to 25 for the proposed combined scheme with a detection probability as high as 0.98. We also examined the averaging collusion scenario, and the improvements for both cases are consistent with the earlier results on synthetic signals.

E. Discussions

1) Security of the Group Information: From the results of the proposed scheme, we can see that the group information helps narrow down the suspicious users in the colluder detection. However, if the group information is not embedded properly, the attackers may figure out the positions of group subcode, and try to frame innocent groups and mislead the detection. Therefore, the embedded group information should have sufficiently high security. In the following, we shall examine the security of the group information for GRACE fingerprinting and compare it with that of the group ECC fingerprinting by appending.

For the group ECC fingerprinting by appending, all of the users inside one group have the same group subcode with length L_g; thus, they have L_g segments in common. On the other hand, for users coming from different groups, their matches in the user subcodes are at most $L - D$, which is usually much smaller than L_g. When several users compare their copies, they can examine the number of the matched segments and figure out whether they belong to one group or not. They may also identify the positions of the group subcode. With the position information of the group subcode, one colluder may contribute his or her share only to the group subcode positions and other colluders from a different group only contribute to user subcode positions. We call this the group-framing attack. Under this attack, after the group detection, the colluder detection will be limited to the group where only one colluder comes from. Since this colluder did not contribute to the user subcode, he or she is less likely to be declared as the colluder. Hence, the probability of accusing an innocent user as a colluder will be high.

For GRACE fingerprinting, each group has a different group subcode from the others. Within one group, users have different user subcodes. As a result of the superposition of these two subcodes, the fingerprint sequence for each user is different from any other user, and the colluders cannot separately identify the group information by comparing their copies. We further note that no matter which segment the colluder contributes, he or she always contributes both the group information and the user information. The group-framing attack mentioned above cannot succeed here. Thus, the joint coding and embedding of GRACE provides both an effective and a secure way to incorporate the group information.

2) Computational Complexity of GRACE Fingerprinting: Compared with the ECC-based fingerprinting, the extra detection computation of the GRACE fingerprinting comes from the detection of guilty groups, which needs $O(gN)$ computations for a total of g groups. Incorporating the computational complexity of the ECC-based fingerprinting derived in Section III-A, the overall computational complexity for the GRACE fingerprinting is $O(qN) + O(gN)$. The group number g is usually much smaller than the total number of users and, in our example, g equals q. Therefore, the overall computational complexity remains at $O(qN)$, the same order as the ECC-based fingerprinting.

It is worth mentioning that since, in most cases, the colluder detection is applied within a small amount of groups, the suspicious user set to be examined will be much smaller than that in nongrouped ECC-based fingerprinting. This further speeds up the colluder detection process.

3) Multilevel GRACE Fingerprinting: The idea of the proposed GRACE fingerprinting is to use the group information to quickly narrow down the suspicious colluders to a small group of users. Within each group, the minimum distance between the users' codewords is larger than that of the whole user set so that the users' codewords are more separated and easier to detect. Following this idea, we can extend our GRACE fingerprinting to general multilevel GRACE fingerprinting to capture more complicated collusion patterns.

For example, we partition a codebook with minimum distance D^0 into groups. Inside each group, the minimum distance D^1 is larger than D^0. Then, we repeat this partition for each group until the minimum distance equals the code length L or the structure of the group can capture the collusion pattern. When combining the group information with the user information, we can adopt a similar strategy used in the tree-based scheme in [3] to assign each level an orthogonal sequence and embed them by proper scaling. At the detector side, the group information at each level is used to narrow down the suspicious colluders to a smaller group, and the colluder can be detected inside the extracted groups as before.

VI. Conclusion

Starting from a cross-layer framework for multimedia fingerprinting, this paper jointly considers the fingerprint encoding, embedding, and detection of ECC-based multimedia fingerprinting. Through examining its performance and comparing it with orthogonal fingerprinting, we have found that the ECC-based fingerprinting has much higher detection efficiency than orthogonal fingerprinting but poorer collusion resistance. In order to improve the collusion resistance of the ECC-based fingerprinting while preserving its efficient detection, we propose two joint-coding-and-embedding techniques, namely, the permuted subsegment embedding technique and the GRACE technique. Our results show the significant performance gain of each approach on the collusion resistance over the conventional ECC-based fingerprinting. We then combine these two new schemes to further improve the collusion resistance and obtain a complete joint-coding-and-embedding design for coded fingerprinting. Our combined design can resist more than three times colluders' collusion as many as that of the conventional ECC-based fingerprinting and retains the low detection computational complexity. It offers a much improved tradeoff between the collusion resistance and detection efficiency than the conventional ECC-based fingerprinting and orthogonal fingerprinting.

References

[1] F. Ergun, J. Kilian, and R. Kumar, "A note on the limits of collusion-resistant watermark," *Lecture Notes Comput. Sci.*, vol. 1592, Jan. 1999.

[2] Z. J. Wang, M. Wu, H. Zhao, W. Trappe, and K. J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 804–821, Jun. 2005.

[3] Z. J. Wang, M. Wu, W. Trappe, and K. J. R. Liu, "Group-oriented fingerprinting for multimedia forensics," *EURASIP J. Appl. Signal Process., Special Issue on Multimedia Security Rights Management*, vol. 2004, no. 14, pp. 2153–2173, Oct. 2004.

[4] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[5] N. R. Wagner, "Fingerprinting," in *Proc. Symp. Security Privacy*, Oakland, CA, Apr. 1983, pp. 18–22.

[6] C. Meadows, G. R. Blakley, and G. B. Purdy, "Fingerprinting long forgiving messages," *Lecture Notes Comput. Sci.*, vol. 218, Jan. 1986.

[7] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, Sep. 1998.

[8] Y. Yacobi, "Improved Boneh-Shaw content fingerprinting," Lecture Notes Comput. Sci. vol. 2020, CT-RSA 2001, 2001, pp. 378–391.

[9] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, Apr. 2003.

[10] D. To, R. Safavi-Naini, and Y. Wang, "A 2-secure code with efficient tracing algorithm," Progress in Cryptology—INDOCRYPT'02 Lecture Notes Comput. Sci., vol. 2551, pp. 149–162, 2002.

[11] A. Barg, G. R. Blakley, and G. Kabatiansky, "Digital fingerprinting codes: problem statements, constructions, identification of traitors," *IEEE Trans. Inform. Theory*, vol. 49, no. 4, pp. 852–865, Apr. 2003.

[12] R. Safavi-Naini and Y. Wang, *Lecture Notes Comput. Sci.*, vol. 2320, pp. 57–75, Jan. 2002.

[13] R. Safavi-Naini and Y. Wang, "Traitor tracing for shortened and corrupted fingerprints," *Lecture Notes Comput. Sci.*, vol. 2696, Jan. 2003.

[14] M. Fernandez and M. Soriano, "Soft-decision tracing in fingerprinted multimedia content," *IEEE Multimedia*, vol. 11, no. 2, pp. 38–46, Apr.–Jun. 2004.

[15] M. Wu, W. Trappe, Z. Wang, and K. J. R. Liu, "Collusion resistant fingerprinting for multimedia," *IEEE Signal Process. Mag.*, vol. 21, no. 2, pp. 15–27, Mar. 2004.

[16] M. Wu and B. Liu, "Data hiding in image and video: part-I—fundamental issues and solutions," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 685–695, Jun. 2003.

[17] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1042–1049, Mar. 2001.

[18] H. V. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting," *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 646–661, May 2005.

[19] S. He and M. Wu, "Performance study of ECC-based collusion-resistant multimedia fingerprinting," in *Proc. 38th CISS*, Princeton, NJ, Mar. 2004, pp. 827–832.

[20] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[21] A. Silverberg, J. Staddon, and J. L. Walker, "Applications of list decoding to tracing traitors," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1312–1318, May 2003.

[22] F. Zane, "Efficient watermark detection and collusion security," FC 2000, Lecture Notes Comput. Sci. 1962 pp. 21–32, 2001.

[23] H. Chu, L. Qiao, and K. Nahrstedt, "A secure multicast protocol with copyright protection," *Proc. ACM SIGCOMM Computer Communications Rev.*, vol. 32, no. 2, Apr. 2002.

[24] S. He and M. Wu, "Improving collusion resistance of error correcting code based multimedia fingerprinting," in *Proc. ICASSP*, Philadelphia, PA, Mar. 2005, pp. 1029–1032.

[25] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 528–538, Aug. 2004.

[26] S. He and M. Wu, "Group-oriented joint coding and embedding technique for multimedia fingerprinting," in *Proc. SPIE Conf. Security, Watermarking Stegonography*, San Jose, CA, Jan. 2005, vol. 5681, pp. 96–105.

[27] T. van Trung and S. Martirosyan, "On a class of traceability codes," *Designs, Codes Cryptogr.*, vol. 31, no. 2, Feb. 2004, pp. 125–132.

Shan He (S'05) received the B.E. and M.S. degrees in automatic control and industrial engineering (Hons.) from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and is currently pursuing the Ph.D. degree in signal processing and communications with the Department of Electrical and Computer Engineering and the Institute of Advanced Computing Studies, University of Maryland, College Park.

She was a Research Intern with Microsoft Research (Redmond, WA) in 2006. Her research interests include information security and multimedia signal processing.

Ms. He received the Best Master Thesis Award from Tsinghua University in 2002 and the Graduate School Fellowship from University of Maryland from 2002 to 2004.

Min Wu (S'95–M'01) received the B.E. degree in electrical engineering (Hons.) and the B.A. degree in economics (Hons.) from Tsinghua University, Beijing, China, in 1996, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 2001.

Currently, she is an Associate Professor in the Department of Electrical and Computer Engineering and the Institute of Advanced Computer Studies, University of Maryland, College Park. Previously, she was with NEC Research Institute and Panasonic Laboratories, Princeton. She co-authored two books *Multimedia Data Hiding* (Springer-Verlag, 2003) and *Multimedia Fingerprinting Forensics for Traitor Tracing* (EURASIP/Hindawi, 2005) and holds five U.S. patents. Her research interests include information security and forensics, multimedia signal processing, and multimedia communications.

Dr. Wu received the National Scicnce Foundation CAREER award in 2002, a University of Maryland George Corcoran Education Award in 2003, a Massachusetts Institute of Technology Technology Review's TR100 Young Innovator Award in 2004, and an ONR Young Investigator Award in 2005. She is a co-recipient of the 2004 EURASIP Best Paper Award and the 2005 IEEE Signal Processing Society Best Paper Award. She is an Associate Editor of IEEE SIGNAL PROCESSING LETTERS, and served as a Guest Editor of a 2004 special issue in *EURASIP Journal on Applied Signal Processing*. She was Publicity Chair of the 2003 IEEE International Conference on Multimedia and Expo.