Building Efficient and Effective Multimedia Fingerprints via Joint Coding and Embedding

Shan He
ECE Department
University of Maryland, College Park
shanhe@eng.umd.edu

Min Wu
ECE Department
University of Maryland, College Park
minwu@eng.umd.edu

ABSTRACT

Digital fingerprinting protects multimedia content from illegal redistribution by uniquely marking every copy of the content distributed to each user. 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 two new techniques, namely, a *Permuted Subsegment Embedding* technique and a *Group-based Joint Coding and Embedding (GRACE)* 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.

1. 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, whereby each user's copy is identified by a unique ID, known as *fingerprint*, embedded in his/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 users combine their copies of the same content to generate a new version. If designed improperly, the fingerprints can be weakened or removed by collusion attacks.

A growing number of techniques have been proposed recently concerning collusion-resistant fingerprinting for multimedia. Many of them fall in one of the two categories, according to whether an explicit discrete coding step is involved. In the non-coded category, a typical example is orthogonal fingerprinting, which assigns each user a spread spectrum sequence as a fingerprint, and the sequence is typically orthogonal to those for other users [1][2]. Non-coded fingerprinting is a natural extension from spread spectrum embedding [3] and is easy to implement. A weakness of non-coded 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. A concept of *marking assumption* was introduced by Boneh and Shaw in [4], and a two-level 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 spread spectrum sequence to embed fingerprint codes into multimedia signals [5]. By explicitly exploiting the multimedia characteristics through selecting appropriate modulation and embedding schemes, a more compact code was introduced in [6] based on combinatorial design to identify colluders through the code bits shared by them. Many recent works on coded fingerprinting [7][8] extend Boneh and Shaw's framework and consider the construction of codes with traceability, such as identifiable parent property (IPP) codes and traceability (TA) codes. Among these codes, TA codes are stronger than other codes in terms of tracing capability and can be systematically constructed using well known error correcting code (ECC). Thus, TA codes are widely used in the coded fingerprinting literature. The authors of [9] applied an ECC-based TA code to multimedia fingerprinting, and extended it to deal with symbol erasures contributed by noise or cropping in multimedia signal domain. 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 are originated from fingerprinting generic data, the special properties and issues of multimedia signal have not been sufficiently explored in the code design. Although some papers [9][10] claimed that their schemes are for multimedia, the embedding issues are handled in a rather abstract level through models based on the marking assumptions [4][9]. 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 modelling 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 have been shown in [6], by jointly exploring embedding and coding, we can substantially limit the effective ways that attackers may exploit. 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 non-coded orthogonal fingerprinting, but its colluder traceability is considerably lower. In order to achieve a better trade-off between collusion resistance and detection computational complexity, we jointly consider coding and embedding during fingerprint design. We propose a *Permuted Subsegment Embedding* technique and a *Group-based Joint Coding and Embedding (GRACE)* technique. The comparisons between the proposed approaches and the existing ECC-based fingerprinting show that the joint coding and embedding fingerprinting strategy substantially improves the collusion resistance of ECC-based fingerprinting, while preserving its advantages of compact representation and efficient detection.

The paper is organized as follows: Section 2 provides a general background on the ECC-based fingerprinting. Section 3 examines the performance of detection efficiency and collusion traceability of the conventional ECC-based fingerprinting. Based on the results obtained from Section 3, we propose a permuted subsegment embedding technique in Section 4 and show its effectiveness through experiments. We present in Section 5 the proposed GRACE technique, along with the design and evaluation of the multimedia fingerprinting system integrating the two proposed techniques. Finally, conclusions are drawn in Section 6.

2. BACKGROUND ON ECC-BASED FINGERPRINTING

A typical framework for coded multimedia fingerprinting includes a code layer and a spread spectrum based embedding layer [11], as shown in Fig. 1. For anti-collusion purposes, we choose a code with tracing capability at the code layer and assign each codeword to one user as the fingerprint. A c-TA code is a widely used code with the property that any colluded version of the codewords by any c colluders has closer distance to at least one of these colluders' codewords than to the innocents' [12]. It can be constructed using an established ECC over an alphabet of size q, as long as the minimum distance D is large enough and satisfies

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

Here, L is the code length, and c is the number of colluders that the code is intended to resist. With the minimum distance achieving the Singleton bound, Reed-Solomon code is a natural choice for constructing c-TA code. Reed-Solomon codes over an alphabet of size q can be used to construct c-TA codes with the codeword number $N_u = q^t$, where $t = \lceil L/c^2 \rceil$. To embed a codeword, we first partition the host signal into L non-overlapped segments, with one segment corresponding to one symbol. We then build q mutually orthogonal spread spectrum sequences $\{\mathbf{w}_j, j = 1, ..., q\}$ with identical energy $||\mathbf{w}||^2$ to represent the q possible symbol values in the alphabet. Each user's fingerprint sequence is constructed by concatenating the spreading sequences corresponding to the symbols in his/her codeword. This fingerprint sequence is then added to the host signal with perceptual scaling to form the ultimate fingerprinted signal.

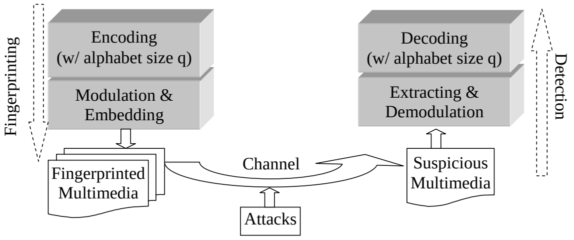


Figure 1: A framework of ECC-based fingerprinting

After the distribution of the fingerprinted copies, users may collaborate and mount cost-effective collusion attacks. The existing works on coded fingerprinting have primarily targeted at code-level collusion resistance. The widely considered collusion model is the *interleaving collusion*, whereby each colluder contributes a non-overlapped set of segments (corresponding to symbols), and these segments are assembled to form a colluded copy. Another major type of collusion is done in the signal domain. A typical example is the *averaging collusion* [2], whereby colluders average the corresponding components in their copies to generate a colluded version. The averaging collusion can be modelled 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. For simplicity in analysis, we assume that the additional noise follows an *i.i.d.* Gaussian distribution. In both types of collusion, the colluders generally make contributions of an approximately equal amount to share the risk of being captured [13].

At the detector side, our goal is to catch one colluder with high probability. We first determine which symbol is present in each multimedia segment through a correlation detector commonly used for spread spectrum embedding [2][3]. As the 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 it to obtain a test signal. Then for each segment of the test signal, we correlate it with each of the q spreading sequences, identify the sequence that gives the maximum correlation, and record the corresponding symbol. The detection statistic for the k^{th} segment is defined as

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

where $\mathbf{z}_k$ and $\mathbf{x}_k$ represent the k^{th} segment of the colluded signal and that of the original signal, respectively. The extracted symbol of k^{th} segment is $\hat{i} = arg \max_{i=1,...,q} T_s(k, i)$. With the sequence of symbols extracted from all media 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) for each of the q possible sequences at each segment without determining the symbol values, and then collect the results from all segments to arrive at the correlation result for each user as

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

where L is the code length, N_u is the total number of users, and the function $sym(j, k)$ is used to retrieve the symbol for the k^{th} segment from the j^{th} user's codeword. Note that this approach has the correlation results equivalent to the matched filter detector that correlates the entire test signal with each user's fingerprint sequence $\mathbf{s}_j$ by

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

Here $\|\mathbf{s}\| = \|\mathbf{s}_j\|$ for all j owing to equal energy construction. The user whose fingerprint has the highest correlation value is identified as the colluder, i.e. $\hat{j} = arg \max_{j=1,...,N_u} T_N(j)$. Compared with the former 2-step hard-decision scheme, the latter scheme takes advantage of the soft information on the symbol level and provides a better colluder-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.

3. PERFORMANCE EVALUATION OF ECC-BASED FINGERPRINTING

Very few of the existing works on ECC-based fingerprinting [5] actually consider the embedding of the designed fingerprints into a host signal and the extraction of them after the collusion. There is little overall performance analysis for ECC-based fingerprinting by jointly considering the coding and embedding, and little comparison with non-coded orthogonal fingerprinting. 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 non-coded orthogonal fingerprinting.

3.1 Efficient Detection and Distribution

For a fingerprinting system with a total of N_u users and a host signal with totally N embeddable components, the detection of the orthogonal fingerprinting is done by correlating the test signal with each user's fingerprint sequence. This takes $N_u N$ multiplications plus $N_u(N-1)$ summations, or a total of $O(N_u N)$ 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 $O(N_u N) + O(N_u) = O(N_u N)$. 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 $N_u L + N_u - 1$ 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 $O(qN)+O(N_u L)$. In many practical applications of robust fingerprinting, to ensure fingerprints be reliably embedded in multimedia, we generally have $N_u << N$. 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 Eqn.(5) with implementation of Eqn.(4) needs $O(qN)$ operations to calculate the partial correlations and further requires $O(N_u L)$ 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 $O(\sqrt[t]{N_u} N)$, which is a substantial improvement over orthogonal fingerprinting especially when the user number gets larger.

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 pre-generate 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/her codeword. To distribute these fingerprinted copies, we can employ secure multicast protocols such as that by Chu et al. [14]. 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 consumption 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 unicast one of the N_u versions of the signals to each user, which generally requires a bandwidth of $N_u B$. When the ECC-based fingerprinting is constructed based on a Reed-Solomon code, for example, with parameters $t = 2, q = 32, N_u = 1024$, 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.

3.2 Collusion Resistance

We measure the collusion resistance of a fingerprinting system in terms of the probability of catching one colluder, denoted as P_d. To get an analytic approximation, we first consider the averaging collusion over an ideal fingerprinting system whose fingerprint sequences have a constant pairwise correlation, denoted as ρ. Without loss of generality, we assume the first c out of n users perform the collusion. The vector of detection statistics T_N's defined in (5) follows an n-dimensional Gaussian distribution:

$$\mathbf{T} = [T_N(1), ..., T_N(n)]^T \sim N([\mathbf{m}_1, \mathbf{m}_2]^T, \sigma_d^2 \Sigma) \qquad (6)$$

$$\text{with } \mathbf{m}_1 = \|\mathbf{s}\|(\frac{1}{c} + (1 - \frac{1}{c})\rho)\mathbf{1}_c, \;\; \mathbf{m}_2 = \|\mathbf{s}\|\rho\mathbf{1}_{n-c}.$$

Here $\mathbf{1}_k$ is an all-1 vector with dimension k-by-1; Σ is an n-by-n matrix whose diagonal elements are 1's, and off-diagonal 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 $\mathbf{m}_1$ and with innocents $\mathbf{m}_2$ are separated more widely for a smaller ρ. This suggests that in absence of any prior knowledge on collusion patterns, a smaller ρ leads to a larger colluder detection probability P_d. We thus prefer fingerprint sequences with a small pairwise correlation ρ when designing a fingerprinting system.

The pairwise correlation of ECC-based fingerprinting can be calculated by examining the code construction. Consider an ECC-based fingerprinting constructed on a Reed-Solomon code with alphabet size q, dimension t, and code length L. Its minimum distance D equals $L - t + 1$. 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

$$\frac{< \mathbf{s}_i, \mathbf{s}_j >}{\|\mathbf{s}\|^2} = \frac{\sum_{k=1}^{L} \mathbf{w}_{ik}\mathbf{w}_{jk}^T}{L\|\mathbf{w}\|^2} \leq \frac{L - D}{L} = \frac{t-1}{L} \triangleq \rho_0. \qquad (7)$$

We can see that codes with a larger minimum distance have a smaller upper bound on the correlation and thus are more preferable. We can choose t and L such that ρ_0 is close to 0. By doing so, the ECC-based fingerprinting and the orthogonal fingerprinting should have comparable resistance against averaging collusion.

To validate the analysis, we examine through simulation the performance of an ECC-based fingerprinting constructed on a Reed-Solomon code with $q = 32$, $t = 2$, $L = 30$, and the number of users $N_u = 1024$. According to the conditions in (1), 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 (7). For comparison purposes, we build orthogonal fingerprinting with the same N_u. Both systems are applied to a host signal that is modelled as an i.i.d. Gaussian sequence with the 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. The detector in (5) is employed for colluder detection. We show the simulation results of P_d for both systems under interleaving and averaging collusion in Fig. 2(a)-(d). The Watermark-to-Noise-Ratio (WNR) ranges from -20dB to 0dB, which includes the scenarios from severe distortion to mild distortion.

From Fig. 2(b)(d) we can see that under averaging collusion, the orthogonal fingerprinting and the ECC-based fingerprinting constructed above have similar colluder identification performance as expected. They both can resist at least a few dozen colluders' averaging attack under high WNR and about half dozen's under very low WNR. However, under interleaving collusion, we observe from Fig. 2(a)(c) a huge gap on the collusion resistance between the two systems. For orthogonal fingerprinting, the probability of colluder detection under interleaving collusion is comparable to that under averaging collusion owing to the orthogonal spreading [6]; at WNR = 0dB, the P_d remains close to 1 when c is around a few dozens. 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. The traceability under interleaving collusion becomes the weak link of the ECC-based fingerprinting and makes it perform much worse than the orthogonal fingerprinting in the collusion resistance.

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 Section 3.1. The significant computational and distribution advantages of ECC-based fingerprinting motivate us to find avenues to improve its collusion resistance and to reduce the performance gap with that of the 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 fingerprints – a *Permuted Subsegment Embedding* technique and a *Group Based Joint Coding and Embedding (GRACE)* technique.

4. PERMUTED SUBSEGMENT EMBEDDING

4.1 The Proposed Embedding Method

The drastic difference in 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 collusion shows that the difference in the resistance against them comes from how we use the embedding layer for collusion resistance. Since each colluded segment comes from just one user, the segment-wise interleaving collusion is equivalent to the symbol-wise interleaving collusion on the code level. The collusion resilience primarily relies on the code layer and almost bypasses the embedding layer. Owing to the limited alphabet size, the chance for the colluders to interleave their symbols and to create a colluded fingerprint close to the fingerprint of an innocent user is very high. On the other hand, for averaging collusion, every colluder contributes his/her share in each segment. Through a correlation detector, the collection of such contributions over the entire test signal leads to high expected correlation values when correlating with the fingerprints from true colluders, and to low expected correlation values when correlating with the fingerprints from innocent users. In other words, the embedding layer helps defending the collusion. This suggests that a closer consideration of the relation between fingerprint encoding, embedding, and detection is helpful to improve the collusion resistance against interleaving collusion.

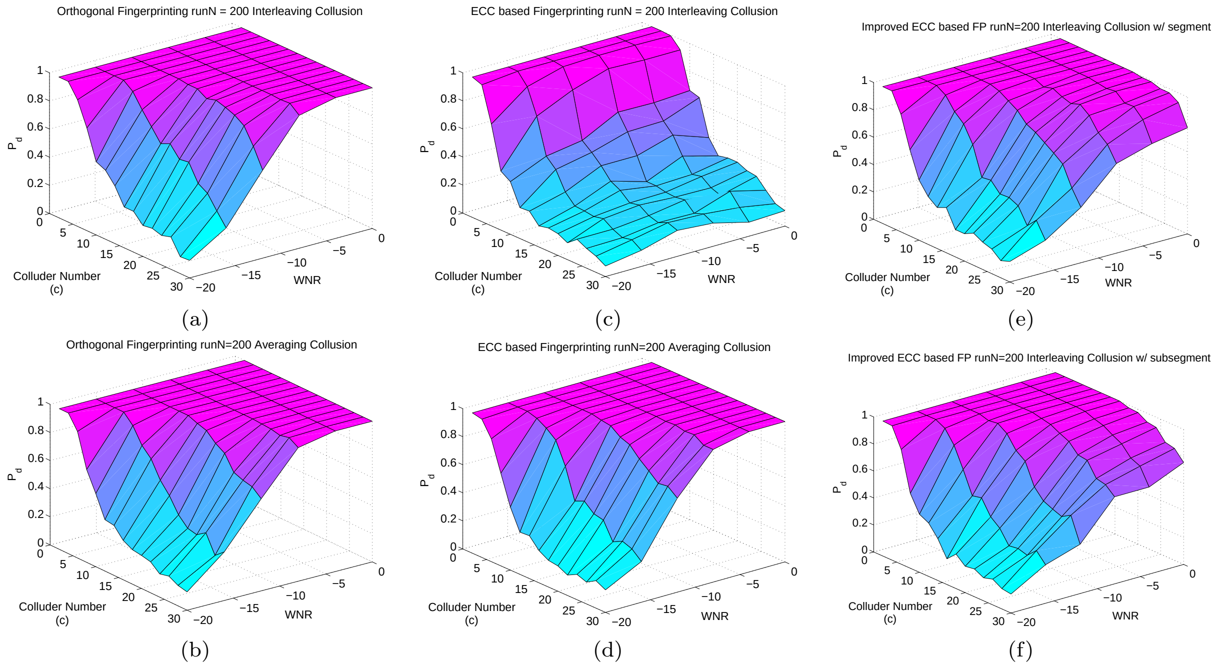


Figure 2: Collusion resistance of orthogonal fingerprinting under (a) interleaving collusion, (b) averaging collusion; ECC-based fingerprinting under (c) interleaving collusion, (d) averaging collusion; Improved ECC-based fingerprinting under (e) segment-wise interleaving collusion, (f) subsegment-wise interleaving.

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 exploiting the code-level limitation. We accomplish this by making each colluded segment contain multiple colluders' contributions. Building upon the existing code construction, we perform two important additional steps that we collectively refer to as *Permuted Subsegment Embedding*. 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 original 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 (5) is applied to the entire fingerprint signal to identify the colluder.

With subsegment partitioning and permutation, each colluded segment after interleaving collusion most likely contains subsegments from multiple users. To the correlation-based detectors, 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.

In the proposed scheme, 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 in the experiment setup in Section 3.2. Our results show that larger β gives higher detection probability P_d, but it may incur higher computational complexity in permutation. Thus a tradeoff should be made when choosing a β value. For the particular system we examined, the improvement of the detection probability saturates when $\beta > 5$. Therefore, we choose $\beta = 5$ for the same system in the experiments to obtain a good trade-off between the permutation computational complexity and the detection performance improvement.

4.2 Experimental Results

We evaluate the performance of the improved system with $\beta = 5$ under various WNRs, and show the results in Fig. 2(e) for segment-wise interleaving collusion. 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. The gap between the performance of the proposed system in Fig. 2(e) and that of the orthogonal fingerprinting in Fig. 2(c) is very small. Fig. 2(f) shows the results for subsegment-wise interleaving collusion. We can see that the proposed system has similar performance under two interleaving collusions and gives a high detection probability for up to two dozen colluders at moderate to high WNRs. Since the permuted subsegment embedding does not affect the performance of the system under averaging collusion, the P_d under averaging collusion remains unchanged.

The above results show that the proposed permuted subsegment embedding provides significant collusion resistance

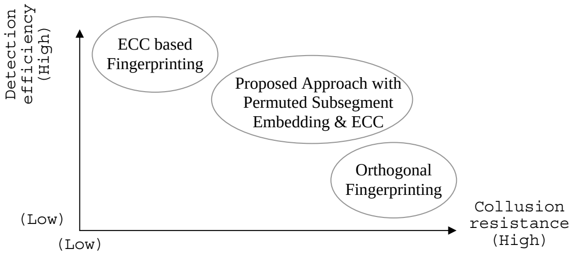

Detection efficiency (High)
ECC based Fingerprinting
Proposed Approach with Permuted Subsegment Embedding & ECC
Orthogonal Fingerprinting
(Low)
(Low)
Collusion resistance (High)

Figure 3: Performance illustration of three fingerprinting designs

improvement for ECC-based fingerprinting. The additional computation cost of the detection comes from the inverse permutation with $O(\beta L)$. Since the maximum value of βL is N, the detection complexity of the improved ECC-based fingerprinting remains at $O(qN)$. The efficient distribution of fingerprinted signal discussed earlier for ECC-based fingerprinting is still applicable here, except that the multicast becomes subsegment based instead of segment based. Moreover, the different user-capacity requirement can be met by preserving the alphabet size but adjusting the dimension of the ECC, such as selecting the t of a Reed-Solomon code, which provides another advantage over orthogonal fingerprinting. We summarize in Fig. 3 the relation of collusion resistance and detection efficiency for three fingerprinting systems, namely, ECC based fingerprinting, improved ECC based fingerprinting with permuted subsegment embedding, and orthogonal fingerprinting. Overall, the improved ECC based fingerprinting provides a better tradeoff among the collusion resistance, detection and distribution efficiency over the conventional schemes. It can accommodate different application requirements through a flexible fingerprinting code construction.

5. GROUP BASED JOINT CODING AND EMBEDDING TECHNIQUE

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 observation, Wang et al. propose group oriented fingerprinting by putting users into groups and adding group information in the fingerprint to enhance the collusion resistance of non-coded orthogonal fingerprinting [16]. This prior knowledge on the collusion pattern has not been exploited for the coded fingerprinting, where new issues arise, such as how to construct groups and how to embed group information. In the meantime, the results in the Section 3.2 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. 2(c) that as WNR increases from -20dB to 0dB, 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 *Group Based Joint Coding and Embedding (GRACE)* fingerprinting system. 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 ECC-based fingerprinting.

5.1 Fingerprint Construction and Embedding

We partition the codewords in ECC-based fingerprinting into groups to capture the collusion pattern, and we 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.

5.1.1 Subcode Construction

To construct the user subcode, we start with a TA code based on ECC construction over an alphabet of size q, as discussed earlier in Section 2. 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 a group have distinct values at each symbol position. Thus the code distance within a group equals the code length L. We assign one codeword to each user as his/her user subcode.

We design the group subcodes to be orthogonal to each other to widely separate the groups and to get accurate group detection. A simple way to construct the group subcode is to use distinct symbols to represent groups; thus, we need a total of g symbols for g groups. For each group, we construct a repetition code with length L by repeating the symbol L times as the group subcode.

5.1.2 Fingerprint Embedding

In the proposed GRACE fingerprinting scheme, we map group subcode and user subcode to two spreading sequences that are orthogonal to each other. We then embed the superposition of these two spreading sequences to the host signal [15]. More specifically, we use the sequences $\{\mathbf{u}_j, j = 1, ..., q\}$ to represent q symbol values in the alphabet of the user subcode, where $\mathbf{u}_j$'s are orthogonal to each other and have identical energy $||\mathbf{u}||^2$. The g sequences $\{\mathbf{a}_i, i = 1, ..., g\}$ represent g group symbols. 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 representing a symbol in the k^{th} segment of user j who belongs to group i as

$$\mathbf{s}_{ijk} = \sqrt{1-\rho}\,\mathbf{u}_{sym(j,k)} + \sqrt{\rho}\,\mathbf{a}_i, \tag{8}$$

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

5.2 Fingerprint Detection

At the detector side, the embedded group information can be used to facilitate the detection by a two-level detection scheme. First, through a correlation detector, we examine the group information in the colluded signal to identify the groups from which the colluders come. More specifically, we subtract the original signal $\mathbf{x}$ from the colluded signal $\mathbf{z}$ to get the test signal, and then extract group information from the test signal using a non-blind correlation detector. The detection statistic with respect to group i is

$$T_G(i) = \frac{(\mathbf{z}-\mathbf{x})^T \mathbf{b}_i}{\|\mathbf{b}\|}, \quad i = 1, 2, ..., g, \tag{9}$$

where $\mathbf{b}_i$ is the concatenation of the spreading sequences representing group i's information from each segment. In the above settings, $\mathbf{b}_i^T = [\mathbf{a}_i^T ... \mathbf{a}_i^T]$ since we embed $\mathbf{a}_i$ in each segment of group i. The k^{th} 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. We then focus our attention on these identified suspicious groups and apply ECC-based fingerprinting detection discussed in Section 2 on the user subcode to narrow down to the true colluders. In this paper, we will employ the soft detector in (5) to correlate the test signal with each user's fingerprint sequence and to identify the one with the highest correlation as the colluder.

5.3 Experimental Results

In this section, we demonstrate the effectiveness of the proposed GRACE fingerprinting through experiments on synthesis signal. 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. Inside each group there are 32 codewords mutually orthogonal to each other. We choose $\rho = 1/7$ in (8) to generate the fingerprint signal from the user subcode and the group subcode. We use the repetition code described in Section 5.1 as the group subcode, and construct *i.i.d.* Gaussian signals with 3×10^4 signal samples to emulate the host signal.

Interleaving collusion is applied to both conventional ECC-based fingerprinting and GRACE fingerprinting. We examine the probability of successfully detecting one colluder (P_d) at WNR = 0dB in the following scenarios:

1) Collusion within a small number of groups: In this case, the grouping correctly reflects the collusion pattern that all the colluders come from a small number of groups. In our simulation, all colluders are from 2 out of 32 groups, and they are randomly distributed between these two groups. The result of P_d under interleaving collusion is shown in Fig. 4(a). We can see that for the same number of colluders, the P_d of the proposed GRACE has 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 certain value e.g. 0.98, the number of colluders the system can resist can be improved from 6 colluders for the conventional ECC-based fingerprinting to 18 colluders for the proposed GRACE 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 result under interleaving collusion is shown in Fig. 4(b), where the proposed GRACE fingerprinting has up to 0.3 improvement on P_d over the conventional ECC-based fingerprinting.

3) Colluders come from distinct groups: In this case, the grouping knowledge is extremely inaccurate. All the colluders come from distinct groups (i.e. the number of groups equals the number of colluders c). The result under interleaving collusion is shown in Fig. 4(c), where the proposed GRACE fingerprinting still has up to 0.2 improvement on P_d over the conventional ECC-based fingerprinting.

The above results can be explained as follows. When collusion happens within a small number of groups, the group information is well preserved so that the group detection of GRACE has high accuracy. As the user subcodes within a small number of groups can be well distinguished because of higher minimum distance than that of the whole codebook, the colluder detection is more accurate than that of the non-group case. When colluders come from multiple groups or even distinct groups and apply interleaving collusion, the energy of the group subcode in 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.

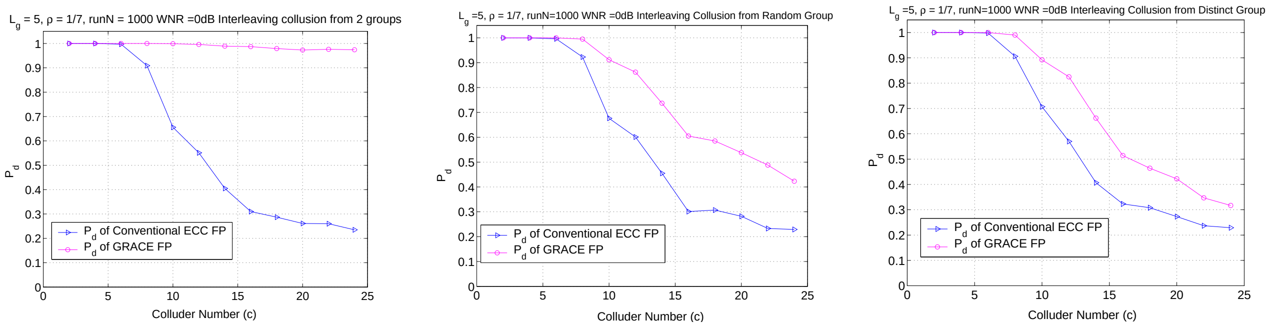
We also examined the cases of averaging collusion and low WNRs. Under averaging collusion, the proposed scheme has similar performance to the conventional ECC-based fingerprinting. At low WNRs, the comparative results are similar to the high WNR case. Overall, the joint coding and embedding as well as the grouping strategy in the proposed GRACE system have brought consistent performance improvement over the conventional ECC-based fingerprinting under various scenarios.

5.4 Combining GRACE with Permuted Subsegment Embedding

Earlier in Section 4, 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 coded fingerprinting system as shown in Fig. 5. We envision that the combined design can provide further improvement on collusion resistance, and we will verify it through experiments.

In the combined design, the fingerprint sequence of 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



(a) 2-Group Interleaving Collusion (b) Random-Group Interleaving Collusion (c) Distinct-Group Interleaving Collusion

Figure 4: Performance comparison of the proposed GRACE fingerprinting and the conventional ECC fingerprinting schemes in terms of probability of detection P_d versus the colluder number c at WNR = 0dB.

two approaches achieves better results than each individual approach. In the cases with inaccurate grouping information (Fig. 6(c)-(f)), the permuted subsegment embedding further 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. 6(a)-(b)), the grouping strategy boosts the detection probability P_d to nearly 1 for a wide range of WNR and c.

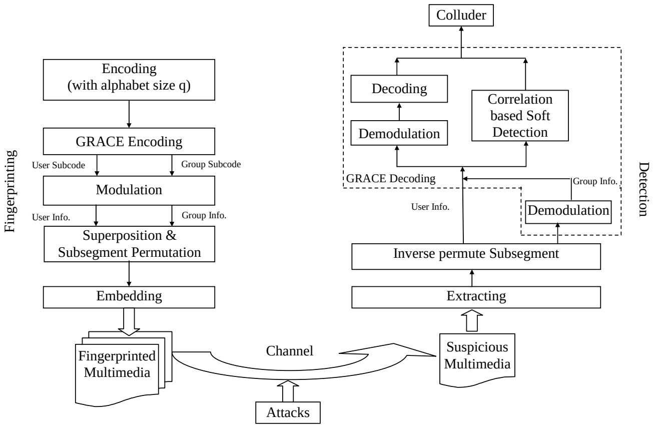


Figure 5: Proposed framework of coded multimedia fingerprinting combining GRACE fingerprinting with permuted subsegment embedding

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 existing ECC-based fingerprinting. We use the transform-domain spread spectrum 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 non-blind detection where the original host signal is available and subtracted from the colluded 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, non-grouped ECC-based fingerprinting scheme, and the other is our proposed GRACE fingerprinting scheme with permuted subsegment embedding. With the same fingerprint coding setup as in Section 3, the effective segment size is 2189 for Lena and 4740 for Baboon. The fingerprinted images have an average PSNR of 41.6dB for Lena and 33.2dB for Baboon. Fig. 7 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 WNR = 0dB. The results of 100 iterations on the two images are shown in Fig. 8, where the number of colluders the system can resist is increased from 6 for the conventional ECC-based fingerprinting to 25 for the proposed combined scheme with detection probability as high as 0.98. The improvement is consistent with the earlier results on synthetic signals.

5.5 Discussions

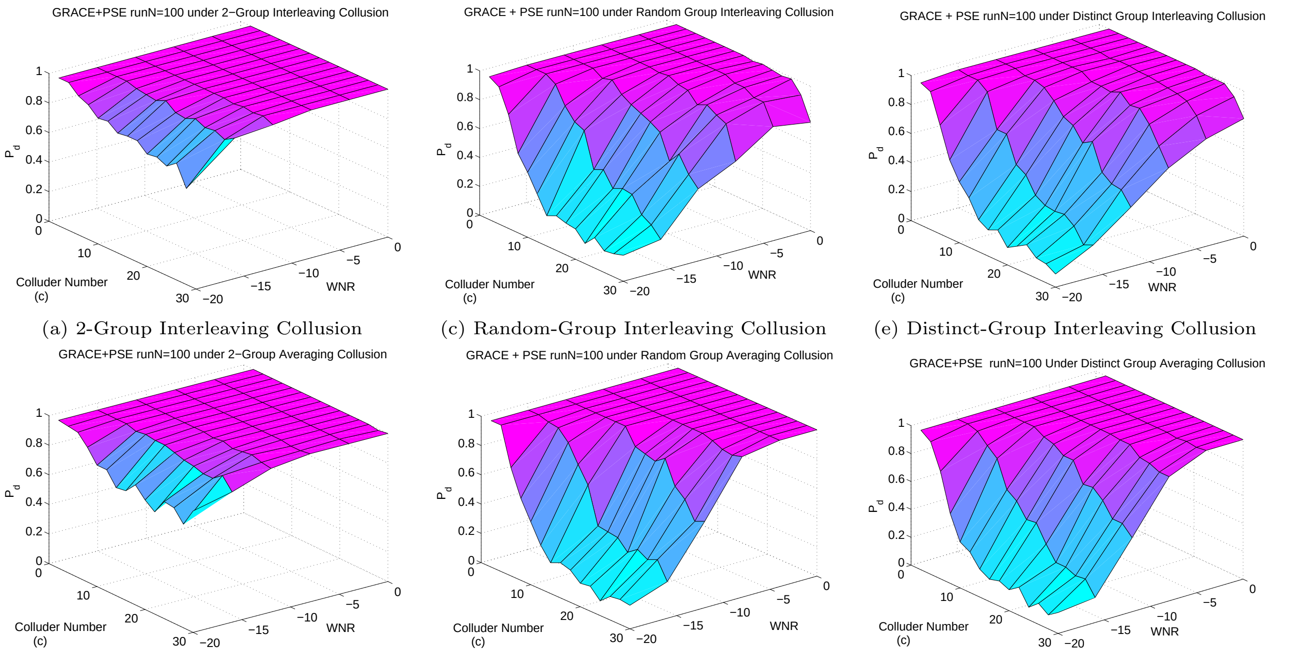
5.5.1 *Computational Complexity of GRACE Fingerprinting*

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 3.1, 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 non-grouped ECC-based fingerprinting. This further speeds up the colluder detection process.

5.5.2 *Multi-level 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


GRACE+PSE runN=100 under 2−Group Interleaving Collusion

GRACE + PSE runN=100 under Random Group Interleaving Collusion

GRACE + PSE runN=100 under Distinct Group Interleaving Collusion

GRACE+PSE runN=100 under 2−Group Averaging Collusion

GRACE + PSE runN=100 under Random Group Averaging Collusion

GRACE+PSE runN=100 Under Distinct Group Averaging Collusion

(a) 2-Group Interleaving Collusion

(c) Random-Group Interleaving Collusion

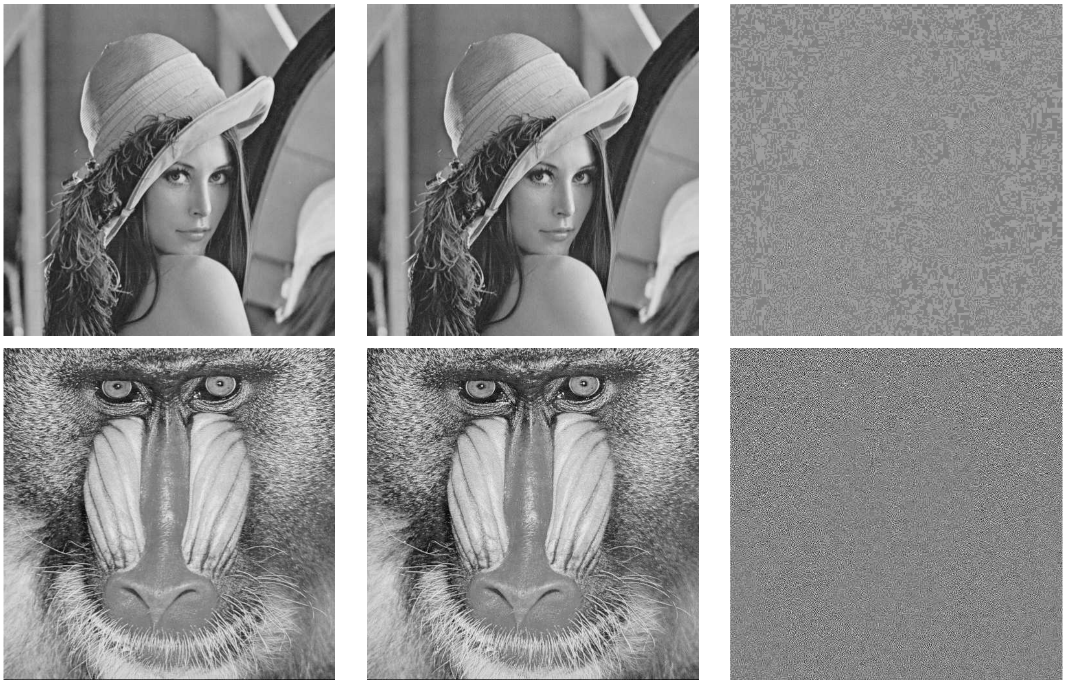
(e) Distinct-Group Interleaving Collusion

(b) 2-Group Averaging Collusion

(d) Random-Group Averaging Collusion

(f) Distinct-Group Averaging Collusion

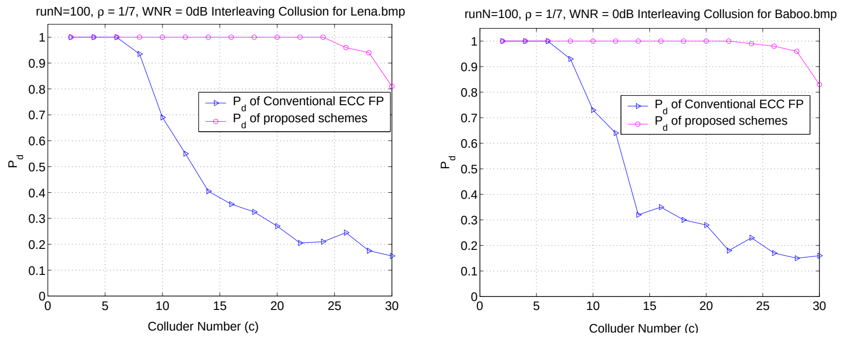
Figure 6: Performance of the proposed GRACE Fingerprinting with permuted subsegment embedding technique: Probability of detection P_d vs. the colluder number c at different WNRs.


(a)

(b)

(c)

Figure 7: (a) Original host images, (b) Fingerprinted images, (c) the corresponding difference images (amplified by a factor of 10).



(a) (b)

Figure 8: Experimental results of the combined scheme for real images of (a) Lena and (b) Baboon under interleaving collusion

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 a general multi-level 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 [16] 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.

6. CONCLUSIONS

Starting from a cross-layer framework of 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 the significant detection efficiency advantage of ECC-based fingerprinting over orthogonal fingerprinting. However, it has poor 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 *Group-Based Joint Coding and Embedding (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 better trade-off between the collusion resistance and detection efficiency than the conventional ECC-based fingerprinting and the orthogonal fingerprinting.

7. REFERENCES

[1] F. Ergun, J. Kilian and R. Kumar, "A Note on the limits of Collusion-Resistant Watermarks", *Eurocrypt '99*, 1999.

[2] Z.J. Wang, M. Wu, H. Zhao, W. Trappe, and K.J.R. Liu, "Collusion Resistance of Multimedia Fingerprinting Using Orthogonal Modulation," *IEEE Trans. on Image Proc.*, to appear in 2005.

[3] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. on Image Processing*, 6(12), pp.1673-1687, 1997.

[4] D. Boneh and J. Shaw, "Collusion-secure Fingerprinting for Digital Data," *IEEE Tran. on Info. Theory*, 44(5), pp. 1897–1905, 1998.

[5] Y. Yacobi, "Improved Boneh-Shaw Content Fingerprinting", *CT-RSA 2001, LNCS 2020*, pp. 378-391, 2001.

[6] W. Trappe, M. Wu, Z.J. Wang, and K.J.R. Liu, "Anti-collusion Fingerprinting for Multimedia", *IEEE Trans. on Sig. Proc.*, 51(4), pp1069-1087, 2003.

[7] D. To, R. Safavi-Naini and Y. Wang, "A 2-secure code with efficient tracing algorithm", *Progress in Cryptology - INDOCRYPT'02, Lecture Notes in Computer Science*, Vol. 2551, Springer-Verlag, pp. 149-162, 2002.

[8] A. Barg, G.R. Blakley and G. Kabatiansky "Digital fingerprinting codes: Problem statements, constructions, identification of traitors" *IEEE Trans. Info. Theory*, 49(4), pp. 852-865, April 2003.

[9] R. Safavi-Naini and Y. Wang, "Collusion Secure q-ary Fingerprinting for Perceptual Content," *Security and Privacy in Digital Rights Management (SPDRM'01)*, pp. 57–75, 2002.

[10] M. Fernandez, and M. Soriano, "Soft-Decision Tracing in Fingerprinted Multimedia Content", *IEEE Multimedia*, Vol.11 No.2, pp38-46, April-June 2004.

[11] S. He and M. Wu, "Performance Study of ECC-based Collusion-resistant Multimedia Fingerprinting," in *Proceedings of the 38th CISS*, March 2004, pp. 827–832.

[12] J.N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial Properties of Frameproof and Traceability Codes", *IEEE Trans. on Info. Theory*, vol. 47, no. 3, pp 1042-1049, 2001.

[13] M. Wu, W. Trappe, Z. Wang, and K.J.R. Liu, "Collusion Resistant Fingerprinting for Multimedia", *IEEE Signal Processing Magazine*,pp 15-27, March, 2004.

[14] H. Chu, L. Qiao, and K. Nahrstedt, "A secure Multicast Protocol with Copyright Protection", *ACM SIGCOMM Computer Communications Review*, vol. 32, no. 2, April 2002.

[15] M. Wu and B. Liu, "Data Hiding in Image and Video: Part-I – Fundamental Issues and Solutions", *IEEE Trans. on Image Proc.*, vol.12, no.6, pp.685-695, June 2003.

[16] Z.J. Wang, M. Wu, W. Trappe, and K.J.R. Liu, "Group-Oriented Fingerprinting for Multimedia Forensics," *EURASIP Journal on Applied Signal Processing*, vol.2004:14, pp.2153-2173, October 2004.