ABSTRACT

Title of Dissertation: A JOINT CODING AND EMBEDDING FRAMEWORK FOR MULTIMEDIA FINGERPRINTING

Shan He, Doctor of Philosophy, 2007

Dissertation directed by: Professor Min Wu
Department of Electrical and Computer Engineering

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. Unauthorized distribution of information has posed serious threats to government and commercial operations. Digital fingerprinting is an emerging technology to protect multimedia content from such illicit redistribution by uniquely marking every copy of the content distributed to each user. One of the most powerful attacks from adversaries is collusion attack where several different fingerprinted copies of the same content are combined together to attenuate or even remove the fingerprints. An ideal fingerprinting system should be able to resist such collusion attacks and also have low embedding and detection computational complexity, and require low transmission bandwidth.

To achieve aforementioned requirements, this thesis presents a joint coding and embedding framework by employing a code layer for efficient fingerprint construction and leveraging the embedding layer to achieve high collusion resistance. Based on this framework, we propose two new joint-coding-embedding techniques, namely, permuted subsegment embedding and group-based joint-coding-embedding fingerprinting. We show that the proposed fingerprinting framework provides an excellent balance between collusion resistance, efficient construction, and efficient detection. The proposed joint coding and embedding techniques allow us to model both coded and non-coded fingerprinting under the same theoretical model, which can be used to provide guidelines of choosing parameters.

Based on the proposed joint coding and embedding techniques, we then consider real-world applications, such as DVD movie mass distribution and cable TV, and develop practical algorithms to fingerprint video in such challenging practical settings as to accommodate more than ten million users and resist hundreds of users' collusion. Our studies show a high potential of joint coding and embedding to meet the needs of real-world large-scale fingerprinting applications. The popularity of the subscription based content services, such as cable TV, inspires us to study the content protection in such scenario where users have access to multiple contents and thus the colluders may pirate multiple movie signals. To address this issue, we exploit the temporal dimension and propose a dynamic fingerprinting scheme that adjusts the fingerprint design based on the detection results of previously pirated signals. We demonstrate the advantages of the proposed dynamic fingerprinting over conventional static fingerprinting. Other issues related to multimedia fingerprinting, such as fingerprinting via QIM embedding, are also discussed in this thesis.

A JOINT CODING AND EMBEDDING FRAMEWORK FOR MULTIMEDIA FINGERPRINTING

by

Shan He

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:

Professor Min Wu, Chair / Advisor
Professor Alexander Barg
Professor K. J. Ray Liu
Professor Gang Qu
Professor Lawrence C. Washington

DEDICATION

To my parents and my husband, Yunfang Feng.

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisor, Prof. Min Wu. Without her support and help, I could never have reached the heights or explored the depths. Her vision and insights have guided me into the amazing research area of multimedia security. Her encouragements have escorted me through many faltering moments. Her patience and endless endeavor have helped me accomplish this dissertation. Her desire and persistence to achieve excellence have inspired and will continue to inspire me to work hard in my professional career.

I appreciate Prof. K. J. Ray Liu for his invaluable advice and continuous guidance during my study, which has constantly propelled me to achieve excellence. I am also indebted to Dr. Darko Kirovski for his encouragement, mentoring, and research support during my internship at Microsoft Research. It was such a wonderful and pleasant experience to work with him and learn from him.

I would like to thank Prof. Alexander Barg, Prof. Gang Qu and Prof. Lawrence C. Washington for serving on my dissertation committee. The knowledge I learned from Prof. Barg's error correcting code course has been very helpful to my research. I am also grateful to Prof. Adrian Papamarcou for his help and support during my graduate studies.

I would also like to thank my friends and colleagues at University of Maryland: Dr. Jane Wang, Dr. Hong Zhao, Dr. Onur Kaya, Dr. Guan-Ming Su, Dr. Yinian Mao, Dr. Yijie Han, Dr. Meng Chen, Hongmei Gou, Ashwin Swaminathan and

Avinash L. Varna. I have enjoyed every moment that we have worked together. I appreciate all their friendship and their help during my graduate study.

Finally, I wish to express my heartfelt gratitude to my family. I thank my mother for raising me with her endless love and for being my friend with tremendous understanding. She is the one who showed me how to become a person with honesty, sincerity and integrity. My father is the person who inspired my interests in engineering ever since I was a child. I would also like to thank my sister and brother for being supportive and considerate during my study. Words fail me to express my appreciation to my husband, Yunfang Feng, whose dedication, love and persistent confidence in me, have made this dissertation possible. I dedicate this dissertation to him and to my parents.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivations

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. Information leak has posed serious threats to commercial markets and government security. According to a survey in 2006 by L.E.K. Consulting LLC under the commission of Motion Picture Association of America (MPAA), U.S. movie studios are losing about $6.1 billion annually in global wholesale revenue to piracy [1]. Another example is that a classified video on Bin Ladens camp shared between the Pentagon and CIA officials was leaked to the news media [2]. Without effective traitor tracing tools, there would remain the reluctance for different agencies to share critical information and thus jeopardize the mission in fighting terrorism and defending the national and global security. Therefore, the protection of multimedia content becomes increasingly important.

Digital fingerprinting is an emerging technology to protect multimedia content from such unauthorized dissemination, whereby a unique ID representing each user, called digital fingerprint, is embedded in his/her copy. When a copy is leaked, the embedded fingerprint can help trace back to the source of the leak. Adversaries may apply various attacks to remove the fingerprints before redistribution. One of the most powerful attacks from attackers is collusion attack, where several different fingerprinted copies of the same content are combined together to attenuate or even remove the fingerprints. In addition to resistance against attacks, three aspects of system efficiency need to be considered when designing an anti-collusion fingerprinting system, namely, the efficiency in constructing, detecting, and distributing fingerprinted signals. Construction efficiency concerns the computational complexity involved during the generation of fingerprinted content; if the complexity is high, we say the construction efficiency is low and vice versa. Similarly, detection efficiency is related to the detection computational complexity. The distribution efficiency refers to the amount of bandwidth consumed during the transmission of all the fingerprinted signals through cable or wireless network. The more bandwidth the transmission requires, the lower the efficiency of distribution is. An ideal fingerprinting system should have high collusion resistance, low embedding and detection computational complexity and low transmission bandwidth.

1.2 Related Prior Work

A growing number of techniques have been proposed recently concerning collusion-resistant fingerprinting for multimedia data. 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 fingerprint and the sequence is typically orthogonal to those for other users [21, 71]. 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 [70]. The non-coded fingerprinting is a natural extension from spread spectrum embedding [17] and is easy to implement. A weakness of non-coded fingerprinting schemes is that the number of long basis sequences needed 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 work can be traced back to the 1980s [7, 69]. A concept of marking assumption was introduced by Boneh and Shaw in [8, 9], 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 spread spectrum sequence to embed fingerprints codes in multimedia signals [78]. By explicitly exploiting the multimedia characteristics through selecting appropriate modulation and embedding schemes, a more compact code was introduced in [64] based on combinatorial design to identify colluders through the code bits shared by them.

Many recent works on coded fingerprinting [4, 5, 62] extend Boneh and Shaw's marking assumption on the collusion 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. For example, the works in [52] and [54] 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 that researchers favor ECC for fingerprint code construction is that some ECCs, such as the algebraic-geometry codes, have efficient decoding algorithms. For example, the Guruswami-Sudan soft-decision list decoding algorithm is employed in [22] for the algebraic-geometry code to identify multiple colluders. In this thesis, we shall refer to the coded fingerprinting constructed on ECC as the *ECC-based Fingerprinting*.

In the existing coded fingerprinting works that 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 [22,52] claimed that their schemes are for multimedia, the embedding issues are handled in a rather abstract level through models based on the marking assumptions [9, 52]. The prior works typically assume that colluders can only change fingerprint symbols in which they have different values, and the colluders assemble pieces of their codes to generate a colluded version. The marking assumption based on generic data are not sufficient to model multimedia fingerprinting where colluders can manipulate fingerprinted multimedia in the signal domain, bringing the equivalent changes in the code domain beyond the conventional marking assumption. In the mean time, as has been shown in [64], 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.

This thesis addresses the issues on the theory and design of collusion-resistant multimedia fingerprinting. We jointly consider coding and embedding and propose two new techniques, which substantially improve the collusion resistance of ECC based fingerprinting, while still preserving its advantages of compact representation and efficient detection. We further extend the joint coding and embedding framework to address the practical applications with challenging requirements of holding millions of users and resisting hundreds of colluders and design fingerprinting schemes for protecting contents in subscription based content services.

1.3 Thesis Organization and Contributions

This dissertation is organized as follows. Chapter 2 starts with background overview on multimedia fingerprinting and then presents a general framework of coded fingerprinting for multimedia signals. Code construction and fingerprint embedding for anti-collusion purposes are discussed. We also examine the performance of conventional ECC-based fingerprinting and compare it with orthogonal fingerprinting, in terms of collusion resistance and detection efficiency.

Based on the framework and the performance examination in Chapter 2, Chapter 3 presents our first proposed joint coding and embedding technique, namely, permuted subsegment embedding technique. We demonstrate the advantages of the proposed technique in terms of collusion resistance, detection complexity and efficient distribution. By employing the proposed technique, we then analyze the collusion resistance of the fingerprinting constructed on different codes, and study the effects of code parameters on the system performance.

Chapter 4 presents the second proposed joint-coding-embedding technique, called group-based joint coding and embedding (GRACE) technique. By taking advantage of the prior knowledge on the collusion pattern, we construct compact fingerprints that consist of user sub-codeword and group sub-codeword and are embedded in host signal via spread spectrum 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. To further improve the detection performance, we propose an adaptive group detection method which can adaptively adjust the group detection parameter according to the observed collusion pattern. We examine the performance of the proposed method and compare it with the existing non-grouped fingerprinting.

Chapter 5 considers how to employ the proposed joint coding and embedding framework and develop practical algorithms to fingerprint video in such challenging practical settings as to accommodate more than ten million users and resisting hundreds of users' collusion. We investigate the proper code structure for large-scale fingerprinting and propose a trimming detection technique that can significantly reduce the detection computational complexity with little reduction in the detection probability. We conduct experiments on video signals to show the potential of joint coding and embedding to meet the needs of real-world large-scale fingerprinting applications.

Chapter 6 explores two research directions related to multimedia fingerprinting. The first one addresses protecting multimedia content from unauthorized redistribution in subscription based services, where adversaries work together to pirate multiple multimedia programs during a subscription period. We exploit the temporal dimension and propose a dynamic fingerprinting scheme that adjusts the fingerprint design based on the detection results of previously pirated signals. We also examine colluders strategies to combat the tracing by dynamic fingerprinting. The second study explores the Quantization Index Modulation (QIM) embedding methods for fingerprinting applications. We first employ Dither Modulation (DM) technique and extend it for embedding multiple symbols through a basic dither sequence design. We then develop a theoretical model and propose a new algorithm to improve the collusion resistance of the basic scheme. We further explore coded fingerprinting based on spread transform dither modulation (STDM) embedding and compare its performance with spread spectrum based fingerprinting under both blind and non-blind detections.

The dissertation is concluded in Chapter 7, with discussions on future perspectives.

Chapter 2

Background and Performance Studies of Multimedia Fingerprinting

In this chapter, we first provide the background for multimedia fingerprinting and briefly discuss fingerprint design schemes. We then introduce a general framework of coded fingerprinting for multimedia signals by integrating coding and embedding issues. Focusing on ECC code construction, we examine the overall performance of conventional ECC-based fingerprinting across both coding and embedding layers, and compare it with orthogonal fingerprinting in various aspects, such as collusion resistance and detection efficiency.

2.1 General Framework of Multimedia Fingerprinting

A multimedia fingerprinting system generally consists of three parts: fingerprint embedding, fingerprint attacks and fingerprint detection. In fingerprint embedding process, content owner generates fingerprint for each user and embed it through robust digital watermarking technique into the original signal to generate each user's copy. The fingerprint should be embedded imperceptibly and robustly [16]. Imperceptibility can be achieved by employing human perceptual model to control the distortion introduced by the embedded fingerprint so that the fingerprinted signal is perceptually similar to the original version. Robustness requires the embedded fingerprints survive intentional or unintentional processing introduced by users or attackers, such as compression and filtering.

After the fingerprinted signals are distributed to end users, adversaries may apply various attacks to try to remove their fingerprints. One powerful and cost-effective attack is called collusion attack, whereby several different fingerprinted copies of the same content are combined together to attenuate or even remove the fingerprints. There are many ways to launch collusion attacks. A simple yet effective way is to average the corresponding signal components or features from multiple copies, called averaging collusion. Attackers can also apply order statistics based collusions such as taking the minimum value of their corresponding signal components. The colluded signal may be distributed outside the authorized group.

Once the content owner obtains the suspicious content, he/she can apply fingerprint detection to extract the fingerprint and then identify the possible colluders. Depending on the availability of the host signal to the detector, fingerprint detection can be performed blindly or non-blindly [16]. In non-blind detection, the host signal is available to the detector and it is usually subtracted from the received signal to remove its interference before detection. In blind detection, the detector does not have access to the host signal and thus its effect cannot be completely removed from the received signal, which may serve as a strong noise to the detector. In most fingerprinting applications, host signal is often available to the detector, and thus in this thesis we mainly focus on non-blind detection. Discussions on fingerprint design under blind detection will be presented in Chapter 6.

2.2 Background on Robust Data Embedding

In this section, we provide an overview of the spread spectrum embedding that has been widely used in the multimedia fingerprinting literature. We then discuss several fingerprint design schemes.

Spread Spectrum Embedding

Due to its excellent robustness and invisibility under non-blind detection [16,29,43], spread spectrum embedding has been widely used in multimedia fingerprinting [34, 37, 71]. As illustrated in Fig. 2.1, watermark is modulated by a random noise-like signal following zero-mean Gaussian distribution, which is then scaled according to human visual model to control the introduced distortion to be lower than the noticeable threshold. The modulated signal is then added to the components of the original signal in the embedding domain to produce the watermarked signal. The embedding process can be formulated as

$$\mathbf{y} = \mathbf{x} + \alpha \cdot JND \cdot \mathbf{s}$$

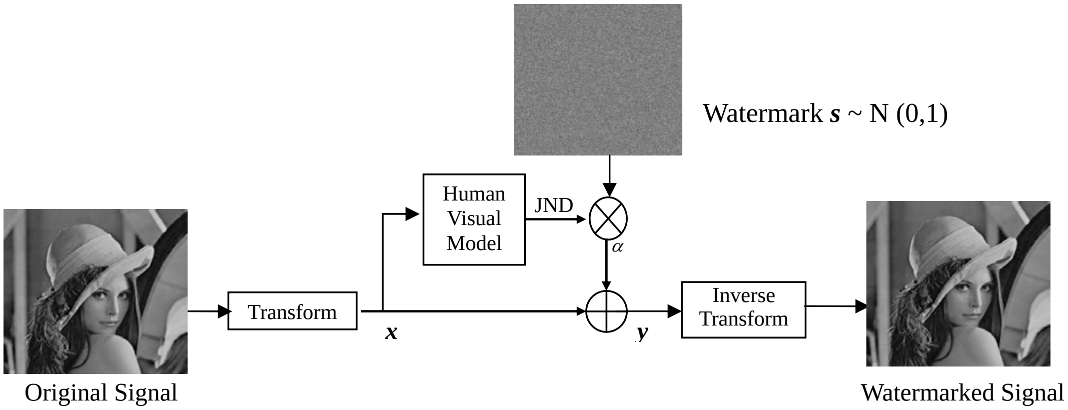


Figure 2.1: Spread Spectrum Watermark Embedding.

where $\mathbf{x}$ is the original signal; $\mathbf{y}$ is the watermarked signal; $\mathbf{s}$ is the watermark; α is the scaling factor used to adjust the watermark energy; and JND denotes *Just Noticeable Difference* and it is employed to control the visual distortion.

During the detection, the host signal $\mathbf{x}$ is first subtracted from the received signal to obtain test signal $\mathbf{z}$. The received signal may have undergone further processings and we model them as additive noise $\mathbf{d}$. The detection of the spread spectrum watermarking can be formulated as a hypothesis testing problem:

$$\begin{cases} H_0: & \mathbf{z} = (\mathbf{x}+\mathbf{d}) - \mathbf{x} = \mathbf{d} \quad \text{watermark is absent,} \\ H_1: & \mathbf{z} = (\mathbf{y}+\mathbf{d}) - \mathbf{x} = \mathbf{s}+\mathbf{d} \quad \text{watermark is present.} \end{cases} \tag{2.1}$$

Under the assumption of additive white Gaussian noise (AWGN), i.e. $\mathbf{d}$ follows *i.i.d.* Gaussian distribution $N(0, \sigma_d^2)$, the optimal detector takes the form of a correlator. The detection statistic T follows a Gaussian distribution

$$T = \mathbf{z}^T\mathbf{s}/\|\mathbf{s}\| = \begin{cases} N(0, \sigma_d^2) & \text{watermark is absent,} \\ N(\|\mathbf{s}\|, \sigma_d^2) & \text{watermark is present.} \end{cases}$$

T is then compared with threshold h: if $T > h$, there is watermark; otherwise, no watermark is detected. The threshold h controls the tradeoff between probability of detection and false alarm.

Fingerprint Design

In general, multimedia fingerprinting can be classified into two categories: non-coded fingerprinting, which does not involve explicit code design during fingerprint construction, and coded fingerprinting, which builds fingerprints based on code structure.

Orthogonal fingerprinting is a typical example of non-coded fingerprinting and is a natural extension from spread spectrum watermarking [71]. In this fingerprinting scheme, each user is assigned a spread spectrum sequence as fingerprint, and sequences for different users are mutually orthogonal. The advantages of orthogonal fingerprinting are that it can well distinguish users and is easy to implement. However, as will be shown in Section 2.4, the required spreading sequences and the detection computational complexity increase linearly with the number of users, which become prohibitively high when the system scales up to hold a large number of users.

Coded fingerprinting introduces correlation among users' fingerprints according to the code structure and thus allows fewer spreading sequences to represent more users. Two types of modulation schemes are usually employed in coded fingerprinting: CDMA modulation and TDMA modulation [74]. In CDMA type of modulation, the fingerprint $\mathbf{s}_j$ for user j is constructed based on a binary code as [64]:

$$\mathbf{s}_j = \sum_{i=1}^{B} b_{ij}\mathbf{u}_i,$$

where b_{ij} is the i-th code bit for user j, which usually takes value from $\{-1, +1\}$,

B is the code length, and $\mathbf{u_i}$ is the spreading sequence for the i-th bit. Through this type of construction, the required base spreading sequences can be reduced to be much fewer than the number of users. However, the computational complexity of fingerprint construction and detection is still on the same order as that of orthogonal fingerprinting and the code is mainly restricted to binary code. On the other hand, as will be shown in this thesis, TDMA type of modulation along with M-ary code provide a lot of freedom to construct fingerprints with fewer spreading sequences and lower computational complexity in detection. We will explore the potential of employing TDMA modulation and ECC code for fingerprint construction. We start with a detailed discussion on ECC-based fingerprinting and performance examination in the following sections.

2.3 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. 2.2, which consists of a coding layer and an embedding layer. As discussed in Section 2.2, the spread spectrum additive embedding technique or its variations is a viable choice for the embedding layer, owing to its excellent robustness under non-blind detection that has been demonstrated in the literature [16–18, 29, 43]. 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 [73, 74]. 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

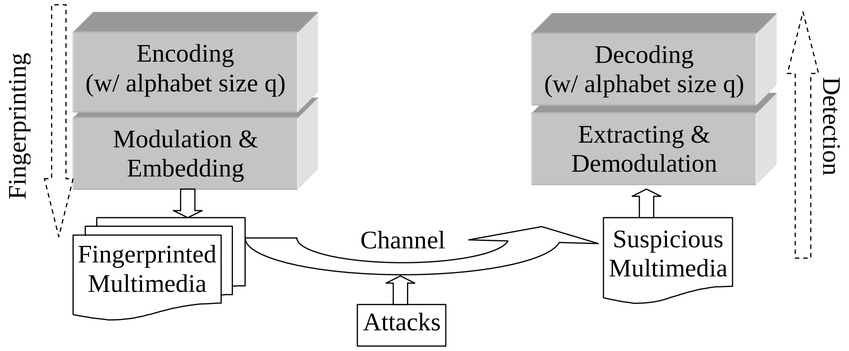


Figure 2.2: A framework of the embedded ECC based fingerprinting.

the marking assumptions [8, 9, 52, 56]. 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 ECC based fingerprinting system includes an ECC code layer and a spread spectrum 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.

2.3.1 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 non-overlapped 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 spread spectrum sequences $\{\mathbf{u}_i, i = 1, ..., q\}$

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 [49]) according to the symbol value in the fingerprint code. Each fingerprinted segment can be modelled as

$$\mathbf{y}_{jk} = \mathbf{u}_{sym(j,k)} + \mathbf{x}_k, \tag{2.2}$$

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

2.3.2 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 [9], and a colluded version is constructed by assembling pieces of the colluders' codewords [52]. 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 [76].

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 [64], bringing changes different from interleaving collusion. The averaging collusion can be modelled as follows:

$$\mathbf{z} = \frac{1}{c} \sum_{j \in S_c} \mathbf{s}_j + \mathbf{x} + \mathbf{d}, \tag{2.3}$$

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 [82] have shown that a number of non-linear 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 thesis. For simplicity in analysis, we assume that the additional noise under both collusions follows *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.

2.3.3 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 spread spectrum embedding [17, 71]. As host signal can be made available to detectors in many fingerprinting applications, we register the suspicious copy with host signal and subtract 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 k^{th} segment is defined as

$$T_s(k,i) = \frac{(\mathbf{z}_k - \mathbf{x}_k)^T \mathbf{u}_i}{\sqrt{\|\mathbf{u}_i\|^2}}, \quad i = 1, 2, ..., q, \tag{2.4}$$

where $\mathbf{z}_k$ and $\mathbf{x}_k$ represent the k^{th} segment of the colluded signal and original signal, respectively. The extracted symbol from k^{th} segment is $\hat{i} = arg \max_{i=1,2,...,q} T_s(k,i)$. 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 Eqn.(2.4) 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

$$T_N(j) = \sum_{k=1}^{L} T_s(k, sym(j,k)) \quad j = 1, 2, ..., N_u, \tag{2.5}$$

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 [51] that correlates the entire test signal with each user's fingerprint sequence $\mathbf{s}_j$ by

$$T_N(j) = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{s}_j}{\sqrt{\|\mathbf{s}\|^2}} \quad j = 1, 2, ..., N_u. \tag{2.6}$$

Here, $\|\mathbf{s}\| = \|\mathbf{s}_j\|$ 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,

[1] As we shall see later in Section 2.4.1, 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 thesis, we shall employ this efficient implementation for the matched-filter detector in Eqn.(2.6) for ECC based fingerprinting.

i.e. $\hat{j} = arg\max_{j=1,2,...,N_u} T_N(j)$. 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 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 non-coded 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.

2.3.4 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 [13, 14] 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 $C = \{\mathbf{v}_1, ..., \mathbf{v}_c\} \subset \Gamma$, where a codeword $\mathbf{v}_i \in \Gamma$ represents the i^{th} colluder and consists of a sequence of L symbols, i.e. $\mathbf{v}_i = [w_1^{(i)} w_2^{(i)} ... w_L^{(i)}]$. A codeword set that can descend from this colluder set is denoted as

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

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

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

for any innocent user's codeword $[s_1 ... s_L] \in \Gamma \diagdown C$, 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 [56]

$$D > (1 - \frac{1}{c^2})L, \tag{2.7}$$

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

As mentioned earlier, most of the existing works [22, 52] 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 is beyond the marking assumptions. The existing work on c-TA codes has been extended to tolerate erasures only [52]. We note that there is nontrivial probability for false alarms in symbol detection, which contributes to the non-erased erroneous colors that are not contributed by any colluders. Thus we need to consider both erasure and non-erasure errors when designing fingerprint code. In the following, we extend the definition of c-TA code to account for both types of errors and develop a minimum distance requirement with a similar strategy used in [52].

As before, we consider a code Γ of length L over an alphabet Σ of size q. This code is called a *c-traceability code tolerating L_e erasures and L_{FA} errors* and denoted as c-$\mathrm{TA}_q(L, N_u; L_{FA}, L_e)$ if for any $(x_1, ..., x_L) \in X(C)$, there is a colluder's codeword $u_i \in C$ such that $|\{j : x_j = w_j^{(i)}\}| > |\{j : x_i = s_j\}|$ for any innocent user's codeword $(s_1, ..., s_L) \in \Gamma \setminus C$. Here $X(C)$ denotes the set of the codeword with no more than L_e erasures and no more than L_{FA} erroneous colors over an extended alphabet $\Sigma \cup \{?\}$, where $\{?\}$ is the erasure symbol. We derive the following minimum distance conditions for c-TA$(L, N_u; L_{FA}, L_e)$ [30]:

Let Γ be an $(L, N_u, D)_q$-ECC, and c an integer. If

$$D > (1 - \frac{1}{c^2})L + \frac{c+1}{c^2}L_{FA} + \frac{1}{c^2}L_e, \tag{2.8}$$

then Γ is c-TA$_q(L, N_u; L_{FA}, L_e)$. The proof can be found in Section 2.6.1.

As can be seen from the above discussions, the ECC based fingerprint code prefers an ECC with larger minimum distance to tolerate more colluders. Among ECC constructions, Reed-Solomon codes have the minimum distance that achieves the Singleton bound [72] and is widely used in the existing coded fingerprinting works [52, 56]. We can construct c-TA code using a Reed-Solomon code that satisfies the above condition. The design can be simplified by treating erasures as errors, which reflects the case of symbol extraction by such schemes as the maximum correlation detector described in Section 2.3.3. This simplification leads to the following results:

Among Reed-Solomon code with alphabet size q, there exists a c-TA$_q(L, N_u; L_{FA})$ code with minimum distance

$$D > (1 - \frac{1}{c^2})L + \frac{c+1}{c^2}L_{FA},$$

where the parameters satisfy

$$N_u = q^t, \quad \text{and} \quad t = \lceil \frac{L}{c^2} - \frac{c+1}{c^2}L_{\text{FA}} \rceil. \tag{2.9}$$

The detailed proof can be found in Section 2.6.1.

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 $c \log N_u$ [26, 44, 55]. 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.

2.4 Performance Analysis of ECC-based Fingerprinting

Examining the existing literature on ECC based fingerprinting reveals that few work has actually taken into full consideration of the embedding layer of the fingerprints. We have found a very limited amount of overall performance analysis by considering the coding and embedding together [78], and little comparison with non-coded 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 non-coded orthogonal fingerprinting.

2.4.1 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 non-coded 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 [79]. 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 $O(qN) + O(N_u L)$. 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.(2.6) with implementation of Eqn.(2.5) 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 $O(\sqrt[t]{N_u} N)$.

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.

2.4.2 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 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. [15]. 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.

2.4.3 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.

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 an additive white Gaussian noise. At the detector side, we consider the soft detector employing matched-filter as in Eqn.(2.6). 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 Eqn.(2.10). For each user, we examine a correlation-based statistic T_N as

$$T_N(j) = \frac{(\mathbf{z}-\mathbf{x})^T\mathbf{s}_j}{\sqrt{\|\mathbf{s}_j\|^2}} \quad j = 1, ..., N_u, \tag{2.10}$$

which follows a multivariate Gaussian distribution of N_u dimensions [50]. 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

$$T_1 = \max_{j \in S_C} T_N(j), \quad T_2 = \max_{j \notin S_C} T_N(j), \tag{2.11}$$

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:

$$\begin{aligned}
m_{T_1} &\triangleq E[T_1] = \frac{\|\mathbf{u}\|}{\sqrt{L}}(\frac{L}{c} + \frac{5(c-1)(L-D)}{12}), \\
\sigma_{T_1}^2 &\triangleq Var[T_1] = \frac{\|\mathbf{u}\|^2}{L}\sigma_C^2 + \sigma_d^2; && (2.12) \\
m_{T_2} &\triangleq E[T_2] = \frac{\|\mathbf{u}\|}{\sqrt{L}} \times \frac{c(L-D)+1}{2}, \\
\sigma_{T_2}^2 &\triangleq Var[T_2] = \frac{\|\mathbf{u}\|^2}{L}\sigma_I^2 + \sigma_d^2; && (2.13) \\
\text{with } \sigma_I^2 &= \left(\frac{c(L-D)-1}{6}\right)^2, \\
\sigma_C^2 &= \left(\frac{5(c-1)(L-D)}{36}\right)^2, && (2.14)
\end{aligned}$$

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

$$P_d = Pr(T_1 > T_2) = \int_{-\infty}^{\infty} P(T_1 > t) f_{T_2}(t)dt, \tag{2.15}$$

where f_{T_2} is the p.d.f. of T_2 and [19]

$$P(T_1 > t) = 1 - Q(\frac{t - m_{T_1}}{\sigma_{T_1}}). \tag{2.16}$$

The detailed derivation is presented in Section 2.6.2.

Averaging Collusion

We employ the matched-filter detector in Eqn.(2.10) 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 Eqn.(2.10) follows an N_u-dimensional Gaussian distribution:

$$\mathbf{T} = [T_N(1), ..., T_N(N_u)]^T \sim N([\mathbf{m}_1, \mathbf{m}_2]^T, \sigma_d^2 \Sigma), \tag{2.17}$$

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

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 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 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 Eqn.(2.7) 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.

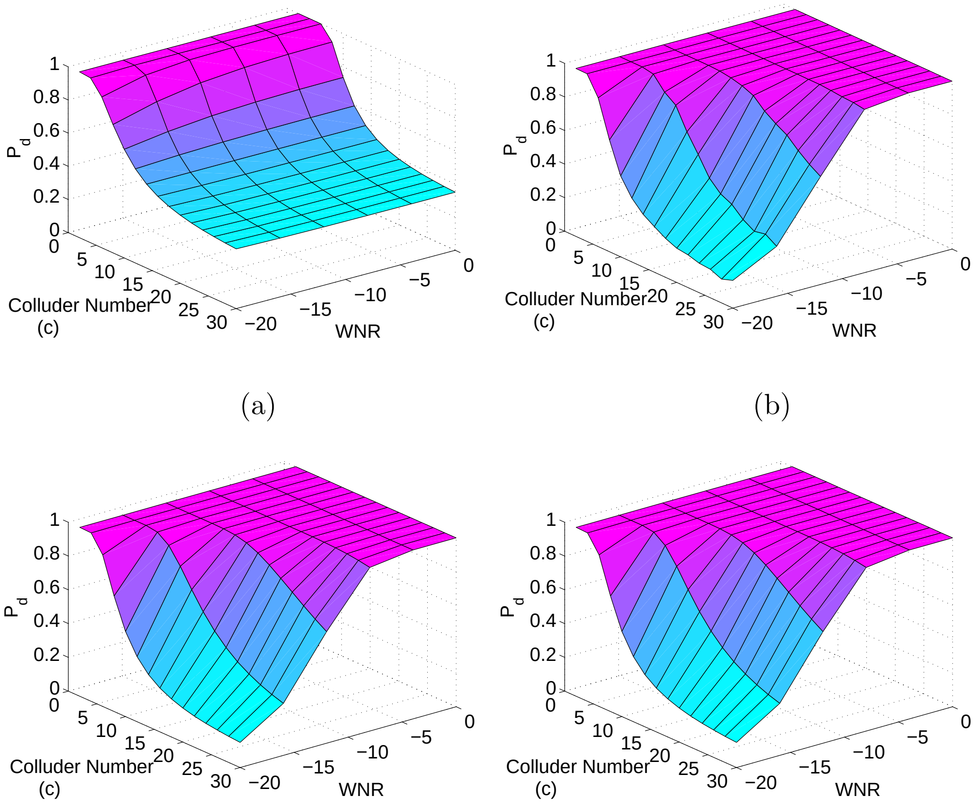
Taking a Reed-Solomon code based fingerprinting as an example, we calculate its pairwise correlation. For an L-tuple q-ary Reed-Solomon code with dimension t, the total number of codewords is $N_u = q^t$ and the minimum distance is $D = L - t + 1$. We use $\mathbf{s}_i$ and $\mathbf{s}_j$ to represent 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

$$\begin{aligned} \frac{< \mathbf{s}_i, \mathbf{s}_j >}{\|\mathbf{s}\|^2} &= \frac{< [\mathbf{w}_{i1}\mathbf{w}_{i2}\cdots\mathbf{w}_{iL}], [\mathbf{w}_{j1}\mathbf{w}_{j2}\cdots\mathbf{w}_{jL}] >}{L\|\mathbf{w}\|^2} \\ &\leq \frac{L-D}{L} = \frac{t-1}{L} = \rho_0. \end{aligned} \tag{2.18}$$

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.

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



(a)

(b)

(c)

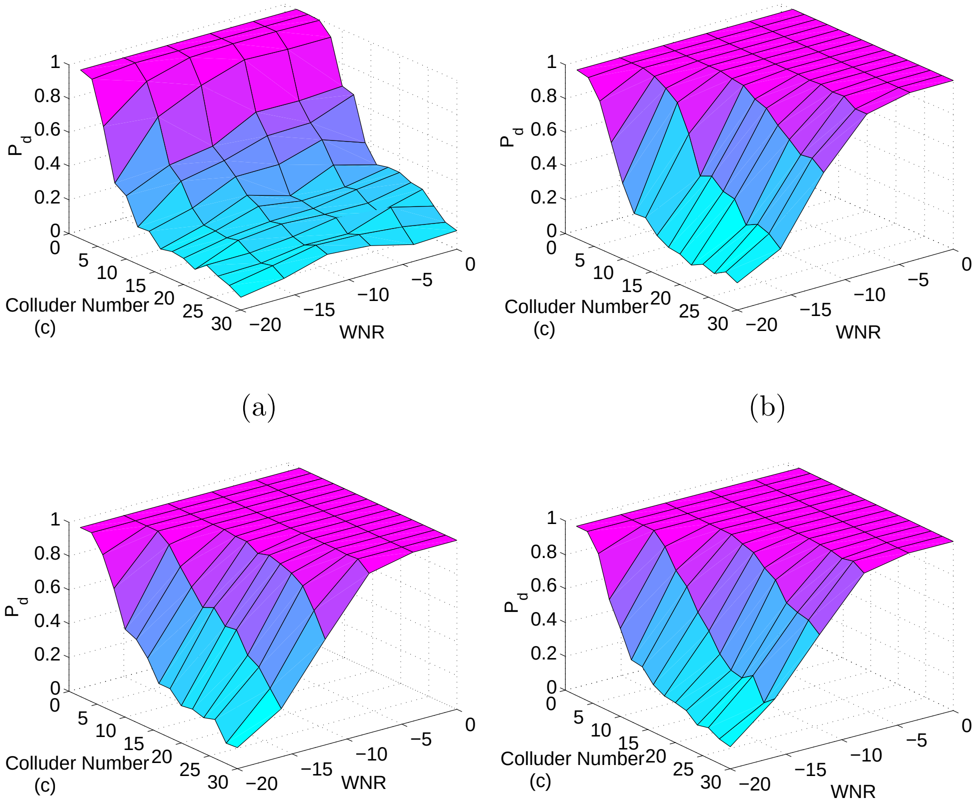
(d)

Figure 2.3: Analytical approximation of ECC-based fingerprinting under (a) interleaving collusion; (b) averaging collusion; and orthogonal fingerprinting under (c) interleaving collusion and (d) averaging collusion.

holding N_u users, the results in Eqn.(2.9) and (2.18) 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 $t = 2$ 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 Eqn.(2.7), 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 Eqn.(2.18), 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. 2.3(a) and (b) respectively. The Watermark-to-Noise-Ratio (WNR) ranges from 0dB to -20dB, which includes the scenarios from severe distortion to mild distortion. The theoretical results for orthogonal fingerprinting from [71] are shown in Fig. 2.3(c) and (d) for interleaving collusion and averaging collusion, respectively. Comparing Fig. 2.3(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 dozens colluders' averaging attack under high WNR and about half dozen's under very low WNR. This is consistent with the above analysis of the collusion resistance against averaging collusion. Thus from colluders' point of view, averaging collusion for an ECC based fingerprinting system is not a very effective strategy. However, under interleaving collusion, we observe from Fig. 2.3(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 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. Thus from colluders' point of view, interleaving collusion is an effective strategy to circumvent the protection.



(a)

(b)

(c)

(d)

Figure 2.4: 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 validate the analysis, we apply both systems to a host signal that is modelled 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 host signal does not have a major effect on the detection performance. The detector in Eqn.(2.10) 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. 2.4. Notice that the analytical approximation of ECC based fingerprinting under interleaving collusion (Fig. 2.3(a)) is higher than the measured value of P_d for large c. This is because the analysis in Eqn.(2.12)-(2.14) 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.

2.5 Chapter Summary

When designing a fingerprinting system, a better trade-off 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 2.4.1 and 2.4.2. 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 two chapters, 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 *Group Based Joint Coding and Embedding (GRACE)* fingerprinting.

2.6 Appendix: Derivations

2.6.1 Derivation of Inequality (2.8) and (2.9)

In this appendix section, we derive the minimum distance requirement of a c-TA$_q(L, N_u; L_e, L_{FA})$ code and the construction of such a code through Reed-Solomon codes.

Let Γ be an ECC of length L with N_u codewords over an alphabet of size q, and c an integer. If its minimum distance D satisfies

$$D > (1 - \frac{1}{c^2})L + \frac{c+1}{c^2}L_{\mathrm{FA}} + \frac{1}{c^2}L_e, \tag{2.19}$$

then Γ is $c - TA_q(L, N_u; L_e, L_{\mathrm{FA}})$.

Proof: Following the strategies by [52], for $\omega \in X(C)$, if $|C| < c$, then there exists $\upsilon_i \in C$ such that υ_i and ω have at least $(L - L_e - L_{\mathrm{FA}})/c$ in common. Here $X(C)$ denotes the set of the codeword with no more than L_e erasures and no more than L_{FA} erroneous colors over an extended alphabet $\Sigma \cup \{?\}$, where $\{?\}$ is the erasure symbol. Then the number of common symbols between υ_i and ω, defined as $\lambda(\upsilon_i, \omega)$, satisfies $\lambda(\upsilon_i, \omega) \geq (L - L_e - L_{\mathrm{FA}})/c$. On the other hand, for any

$\varphi \in \Gamma \setminus C$,

$$\lambda(\varphi, \omega) \leq \lambda(\varphi, \upsilon_1) + ... + \lambda(\varphi, \upsilon_c) + L_{\text{FA}} \leq c\lambda_{max} + L_{\text{FA}}.$$

From (2.7), we have $c^2(L-D) + cL_{\text{FA}} < L - L_e - L_{\text{FA}}$. It follows that $\lambda(\varphi, \omega) \leq c\lambda_{max} + L_{\text{FA}} \leq \lambda(\upsilon_i, \omega)$. Thus υ_i can be identified correctly. □

For simplicity, we can treat the erasures as errors i.e. $L_e = 0$, which reflects the case of the maximum detector we considered in this thesis. Then we have the following corollary for constructing $c - \text{TA}_q(L, N_u; L_{\text{FA}})$ code via Reed-Solomon code.

Among Reed-Solomon code with alphabet size q, there exists a $c-\text{TA}_q(L, N_u; L_{\text{FA}})$ code with minimum distance

$$D > (1 - \frac{1}{c^2})L + \frac{c+1}{c^2}L_{\text{FA}}, \tag{2.20}$$

where the total codeword number $N_u = q^t$ and $t = \lceil \frac{L}{c^2} - \frac{c+1}{c^2}L_{\text{FA}} \rceil$.

Proof: The minimum distance for a Reed-Solomon code is $D = L - t + 1$. If we choose

$$t = \lceil \frac{L}{c^2} - \frac{c+1}{c^2}L_{\text{FA}} \rceil,$$

then

$$t < \frac{L}{c^2} - \frac{c+1}{c^2}L_{\text{FA}} + 1.$$

Therefore,

$$D = L - t + 1 > (1 - \frac{1}{c^2})L + \frac{c+1}{c^2}L_{\text{FA}}.$$

From Eqn. (2.19), this code is a $c - \text{TA}_q(L, N_u; L_{\text{FA}})$ code. □

2.6.2 Derivation of Eqns. (2.12)-(2.14)

In this appendix section, we presents the detailed derivation of the detection statistic distribution under interleaving collusion.

Given the colluded signal $\mathbf{z}$ generated through interleaving followed by additive white gaussian noise, we obtain the correlation-based statistic T_N for each user as

$$T_N(j) = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{s}_j}{\sqrt{\|\mathbf{s}_j\|}} \quad j = 1, ..., N_u, \tag{2.21}$$

where $\mathbf{x}$ is the host signal, $\mathbf{s}_j$ is the fingerprint signal of user j by concatenating the fingerprint sequence corresponding to the symbols in user j's codeword. $T_N(j)$ follows a Gaussian distribution

$$T_N(j) \sim N(\frac{\|\mathbf{u}\|}{L} \times (L - d(r, c_j)), \sigma_d^2). \tag{2.22}$$

Here, r is the extracted colluded codeword, c_j is the codeword for user j, $d(\cdot, \cdot)$ is the hamming distance metric, and $\|\mathbf{u}\|$ is the strength of the fingerprint sequence corresponding to one symbol. From the modulation and embedding layer, all the $\|\mathbf{s}_j\|$'s are the same and equal to $\sqrt{L}\|\mathbf{u}\|$ with the code length denoted as L. We define

$$T_1 = \max_{j \in S_C} T_N(j), \quad T_2 = \max_{j \notin S_C} T_N(j), \tag{2.23}$$

where S_C is the colluder set.

For simplicity, we approximate T_1 and T_2 using Gaussian distributions conditional on random variables M_C and M_I, respectively

$$T_1 | M_C \sim N(\frac{\|\mathbf{u}\|}{L} M_C, \sigma_d^2), \quad T_2 | M_I \sim N(\frac{\|\mathbf{u}\|}{L} M_I, \sigma_d^2), \tag{2.24}$$

where $M_C = \max_{j \in S_C}(L - d(r, c_j))$ and $M_I = \max_{j \notin S_C}(L - d(r, c_j))$ indicate the maximum number of matched symbols of colluded codeword with colluders' codewords and with innocent users' codewords respectively. Notice that for M_I, the maximum value is $c(L-D)$ and minimum value is 1. Then we approximate the mean and variance of M_I as $(c(L-D)+1)/2$ and $\sigma_I^2 = ((c(L-D)-1)/6)^2$, respectively. Similarly, the maximum value of M_C is $L/c+(c-1)(L-D)$ and minimum value is L/c and we approximate its mean as $L/c+\beta(c-1)(L-D)$ and variance as $\sigma_C^2 = (\beta(c-1)(L-D)/3)^2$. β is set to a value less than 1/2, which reflects the fact that lower values of the M_C is more likely to happen than higher values. Under these assumptions, we can further approximate T_1 and T_2 using Gaussian distribution with means and variances calculated as follows:

$$\begin{aligned}
m_1 &= E[T_1] = E_{M_C}[E[T_1|M_C]] = \frac{\|\mathbf{u}\|}{\sqrt{L}}(\frac{L}{c}+(c-1)(L-D)\beta) \\
\sigma_1^2 &= Var[T_1] = \frac{\|\mathbf{u}\|^2}{L}\sigma_C^2+\sigma_d^2 \\
m_2 &= E[T_2] = E_{M_I}[E[T_2|M_I]] = \frac{\|\mathbf{u}\|}{\sqrt{L}} \times \frac{c(L-D)+1}{2} \\
\sigma_2^2 &= Var[T_2] = \frac{\|\mathbf{u}\|^2}{L}\sigma_I^2+\sigma_d^2 \\
&\text{with} \quad \sigma_I^2 = (\frac{c(L-D)-1}{6})^2, \quad \sigma_C^2 = (\frac{\beta(c-1)(L-D)}{3})^2
\end{aligned}$$

where β is used to adjust the approximation. We examined several β values, and choose $\beta = 5/12$ for a good approximation.

Chapter 3

Joint Coding and Embedding with Permuted Subsegment Embedding

From Chapter 2, we have seen that coded fingerprinting has efficient detection but rather low collusion resistance. In this Chapter, we explore avenues that can both retain the advantages provided by the ECC-based fingerprinting and improve the collusion resistance. We have observed that the existing ECC fingerprinting works put most of the attention on the code layer and few work has considered the interaction between coding and embedding. In the mean time, joint consideration of coding and embedding has shown promising results recently in [64] for non-segment based fingerprinting. This motivates us to examine the interplay between the ECC code layer and the embedding layer. As we shall see, by employing a strategic embedding mechanism referred to as the *Permuted Subsegment Embedding* for putting the ECC fingerprint code into host media, we can benefit from the joint consideration of coding and embedding for ECC-based fingerprinting and substantially improve its collusion resistance. Based on the proposed embedding technique, we will further study how to choose fingerprinting codes to meet various requirements on collusion resistance and detection efficiency.

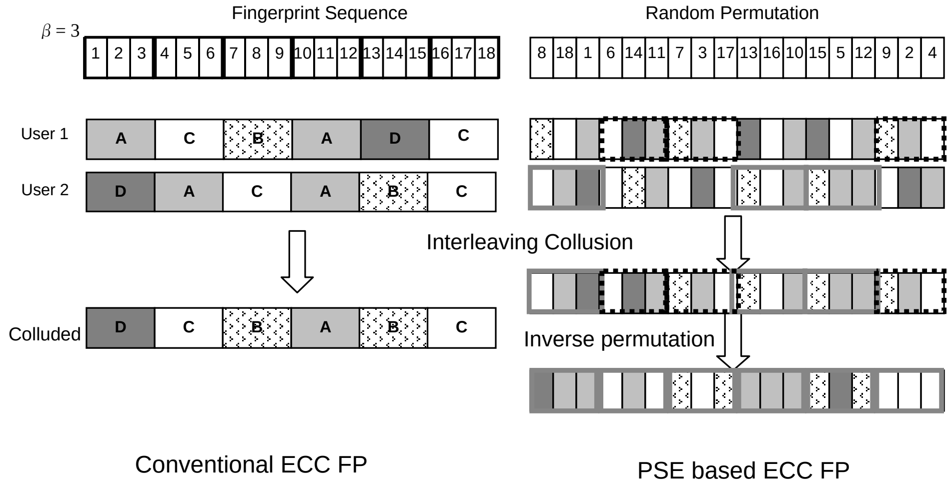
3.1 Permuted Subsegment Embedding

3.1.1 The Proposed Embedding Method

We have observed from Section 2.4.3 a drastic difference in the collusion resistance against averaging and interleaving collusions of ECC based fingerprinting. This 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 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 to 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 averaging collusion, every colluder contributes his/her share in every segment. Through a correlation detector, the collection of such 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* [32]. As shown in Fig. 3.1, 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 Eqn.(2.10) is applied to identify the colluder.

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



(a) (b)

Figure 3.1: Illustration of the permutated subsegment embedding for ECC based fingerprinting: (a) the conventional ECC based fingerprinting, (b) the proposed scheme.

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.

3.1.2 Detection Analysis against Interleaving Collusion

Consider an ECC-based fingerprinting on a code $[L, t, D]$ with code length L, dimension t and minimum distance D. Each symbol is mapped to a spreading sequence with strength $\|\mathbf{u}\|$. After the permuted subsegment embedding with β, each fingerprint sequence consists of βL subsegments, and each subsegment has strength $\|\mathbf{u}\|/\beta$. Notice that after permuted subsegment embedding, the colluders cannot identify which subsegment corresponds to which symbol even if they know the segment size. Thus, permuted subsegment embedding has the effect that it forces colluders to perform interleaving collusion using subsegments. In the following analysis, we take this observation and approximate the collusion strategy as subsegment-wised interleaving collusion, under which, the permutation does not affect the results.

Denote the colluded fingerprint sequence as $\mathbf{y}$ which is generated by subsegment-wise interleaving from c colluders plus an additive Gaussian noise $\mathbf{n}$ with mean 0 and variance σ_d, i.e.

$$\mathbf{y} = ITL(s_1, s_2, ..., s_c) + \mathbf{n}, \tag{3.1}$$

where ITL denotes the subsegment-wise interleaving collusion. Without loss of generality, we assume the first c colluders perform collusion attack.

We employ correlation based detection as we mentioned earlier in Chapter 2, i.e.

$$T_N(j) = \frac{(\mathbf{z}-\mathbf{x})^T \mathbf{s}_j}{\sqrt{\|\mathbf{s}_j\|^2}} \quad j = 1, ..., N_u. \tag{3.2}$$

T_N follows a Gaussian distribution for both colluders and innocent users with different mean values, i.e.,

$$\mathbf{T} = N([\mathbf{m}_1, \mathbf{m}_2], \Sigma) \tag{3.3}$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ are the mean values for colluder and innocent user, respectively, and they are determined by the matches between the colluded signal (before applying additive noise) and colluders'(innocents') users. In a fair subsegment interleaving collusion, the colluders contribute approximately equal amount of subsegments and thus one colluder contributes $\lfloor \beta L/c \rfloor$ subsegments on average. Besides the contributed segments, matches may also come from the shared segments among the codewords. For a $[L, t, D]$ code, there are maximum of $L - D$ matches between codewords for L symbols. Then for $(\beta L - \lfloor \beta L/c \rfloor)$ symbols, there are $\lfloor(\beta L - \lfloor\beta L/c\rfloor)(L-D)/L\rfloor$ matches on average. Thus we arrive at $\lfloor\beta L/c\rfloor + \lfloor(L-D)/L(\beta L - \lfloor\beta L/c\rfloor)\rfloor$ matches between the colluded codeword and user i's codeword. For an innocent user j, the matches only come from the code itself, it would be $\lfloor((L-D)/L)L\beta\rfloor$.

The variance matrix Σ can be derived as follows:

$$\begin{aligned}
\Sigma(i,j) &= E[(T_N(i) - E(T_N(i)))(T_N(j) - E(T_N(j)))] \\
&= E[\frac{(\mathbf{n}^T\mathbf{s}_i)(\mathbf{n}^T\mathbf{s}_j)}{\|s\|^2}] = \mathbf{s}_i^T E[\mathbf{n}\mathbf{n}^T]\mathbf{s}_j = \mathbf{s}_i^T\mathbf{s}_j \qquad (3.4) \\
&= \begin{cases} \frac{L-D}{L} & \text{for} \quad i \neq j \\ 1 \quad \text{for} \quad i = j \end{cases} . \qquad (3.5)
\end{aligned}$$

In summary, the detection statistic T_N follows N_u dimension Gaussian distribution:

$$\begin{aligned}
\mathbf{T} &\sim N([\mathbf{m}_1^{ITL}, \mathbf{m}_2^{ITL}], \Sigma) \qquad (3.6) \\
\text{with} \quad \mathbf{m}_1^{ITL} &= (\lfloor\frac{\beta L}{c}\rfloor + \lfloor\frac{L-D}{L}(\beta L - \lfloor\frac{\beta L}{c}\rfloor)\rfloor)\frac{\|\mathbf{s}\|}{\beta L}; \qquad (3.7) \\
\mathbf{m}_2^{ITL} &= \lfloor\frac{L-D}{L}L\beta\rfloor\frac{\|\mathbf{s}\|}{\beta L}; \qquad (3.8)
\end{aligned}$$

Recall that under averaging collusion, the detection statistic follows the distribution

$$\mathbf{T} = [T_N(1), ..., T_N(N_u)]^T \sim N([\mathbf{m}_1^{AVG}, \mathbf{m}_2^{AVG}]^T, \sigma_d^2\Sigma), \qquad (3.9)$$
$$\text{with} \quad \mathbf{m}_1^{AVG} = \|\mathbf{s}\|(\frac{1}{c} + (1 - \frac{1}{c})\rho)\mathbf{1}_c, \quad \mathbf{m}_2^{AVG} = \|\mathbf{s}\|\rho\mathbf{1}_{N_u - c},$$

where $\rho = \frac{L-D}{L}$. We can see that as β increase, T_N approaches the distribution under averaging collusion. This can be shown through the following example. We take the Reed-Solomon Code $[L, t, D]$ = [30, 2, 29] and consider $c = 25$. Under averaging collusion, $\mathbf{m}_1^{AVG} = 2.16\|s\|/30 \cdot \mathbf{1}_c, \mathbf{m}_2^{AVG} = \|s\|/30 \cdot \mathbf{1}_{N_u-c}$. Under interleaving collusion, $\mathbf{m}_2^{ITL} = \|s\|/30 \cdot \mathbf{1}_{N_u-c}$ for any β value, which is the same as that under averaging collusion. We denote $\mathbf{m}_1^{ITL} = A\|s\|/L \cdot \mathbf{1}_c$. Then it is sufficient to examine the coefficient A under different β values and compare it with 2.16. For example, when $\beta = 1$, it is the traditional ECC fingerprinting $A = (\lfloor \frac{30}{25} \rfloor + \lfloor \frac{1}{30}(30 - \lfloor \frac{30}{25} \rfloor) \rfloor)/\beta = 1$. We summarize the results for other β values in Table 3.1.

Table 3.1: Coefficient A values for different β

β	1	2	4	5	10	...	1000
A	1	1.5	1.75	2	2.1	...	2.16

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 2.4.3. That is, we choose Reed-Solomon code of length 30, dimension 2 and minimum distance 29 for fingerprint construction. The host signal length is 10,000. Fig. 3.2 shows the results when a total of $c = 25$ colluders perform segment-wise interleaving with WNR = 0dB. 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. 3.2, the improvement on the detection probability saturates when $\beta > 5$. Therefore, we choose $\beta = 5$ for this system in later experiments to

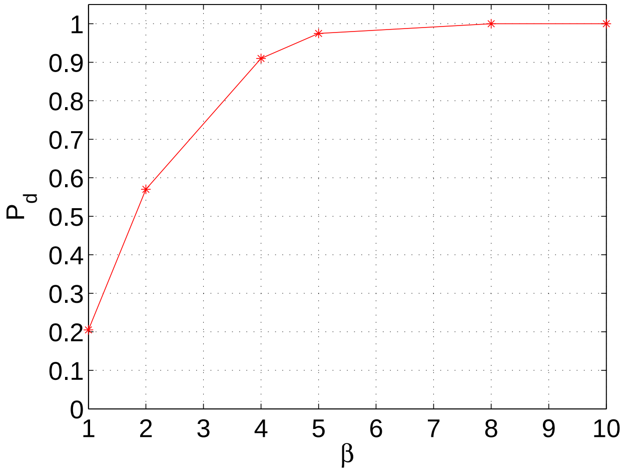


Figure 3.2: Probability of catching one colluder P_d versus β for $c = 25$ and WNR $= 0$dB of the proposed scheme.

obtain a good trade-off between the permutation computational complexity and the detection performance improvement.

3.1.3 Experimental Results

We evaluate the performance of the improved system with $\beta = 5$ under various WNRs, and show the results in Fig. 3.3(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. 3.3(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 to up to four times of that for the conventional ECC based fingerprinting at high and moderate WNRs. In the meantime, the gap between the performance of the proposed system in Fig. 3.3(a) and that of the orthogonal fingerprinting in

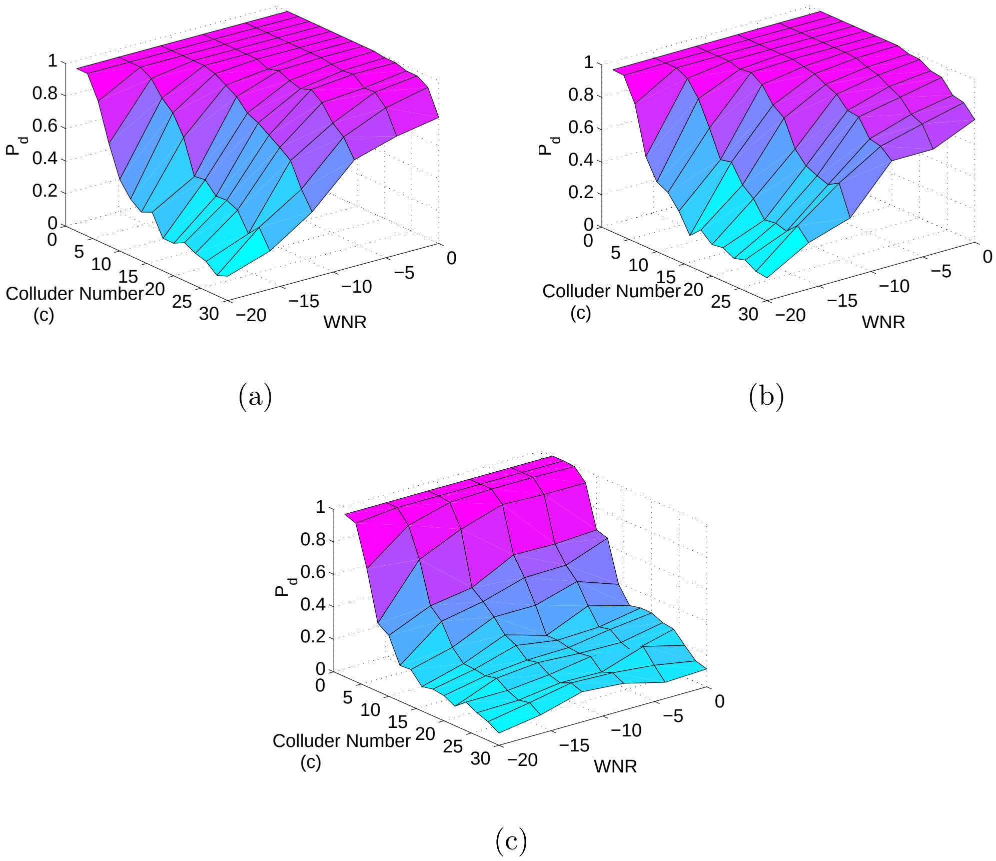
Fig. 2.4(c) is very small.

Next, Fig. 3.3(b) shows the results for interleaving collusion using subsegment as a unit. We observe from Fig. 3.3(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 have 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 dozens 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.

3.1.4 Discussions

The 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 [75], and was applied to a simple staircase construction of binary fingerprint code to prevent framing innocent users [9]. In our proposed work,



(a)

(b)

(c)

Figure 3.3: 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.

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.

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 2.4.1, the other two steps need at most $O(qN)$ computations. Thus the improved ECC fingerprinting has 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 fingerprinted signal discussed earlier for 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 [48]. 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.

Comparison Criteria

The results in Fig. 3.3 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 dimension parameter t. We summarize in Table 3.2 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.

Table 3.2: Performance Comparison of Fingerprinting Systems

	Orthogonal FP	Improved ECC FP	ECC FP
Collusion Resistance to interleaving collusion (colluder #)	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 (colluder #)	Same as interleaving collusion	Similar to Orth FP for small ρ	Similar to Orth FP for small ρ
Detection Computational Complexity	$O(N_u N)$	$O(\sqrt[t]{N_u} N)$	$O(\sqrt[t]{N_u} N)$
Distr. Efficiency γ	1	q^{1-t}	q^{1-t}

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. 3.3 to the corresponding MSE distortion.

3.2 The Effect of Code Parameters on Fingerprinting Performance

We have seen from the previous section that the proposed permuted subsegment embedding substantially improves the collusion resistance of coded fingerprinting. With this improvement, coded fingerprinting has a better trade-off between collusion resistance and detection efficiency than the non-coded fingerprinting. One question that remains to be answered is the effect of the code parameters on the performance of the fingerprinting systems. In this section, building upon the cross-layer framework and employing our proposed permuted subsegment embedding technique, we examine the performance of different fingerprint codes, that have tracing capability and are able to resist collusion. We collectively call these codes traceability codes. The term of "traceability codes", as will be discussed later, also refers to a specific type of traceability codes with the property that the colluded codeword has smaller distance to one of the guilty codewords than any other innocent codeword. To avoid confusion, in this section, we will use "TA codes" to represent this type of traceability codes.

3.2.1 Traceability Codes

In the literatures of fingerprint code design, codes such as Identifiable Parent Property(IPP) codes and Traceability(TA) codes are widely studied [5, 13, 14, 52, 56, 65, 66]. We briefly review these two kinds of codes in the following.

c-TA Code

A c-TA code satisfies the condition that any colluded codeword by any c (or fewer) colluders has a smaller distance to at least one of these colluders' codewords than to the innocent users' [56]. We can construct a c-TA code using an established Error Correcting Code (ECC), provided that the minimum distance D is large enough and satisfies [56]

$$D > \left(1 - \frac{1}{c^2}\right) L. \tag{3.10}$$

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, a Reed-Solomon code is a natural choice for constructing a c-TA code. Then, the number of c-TA codewords over an alphabet of size q constructed through a Reed-Solomon code is $N_u = q^t$, where $t = \lceil L/c^2 \rceil$.

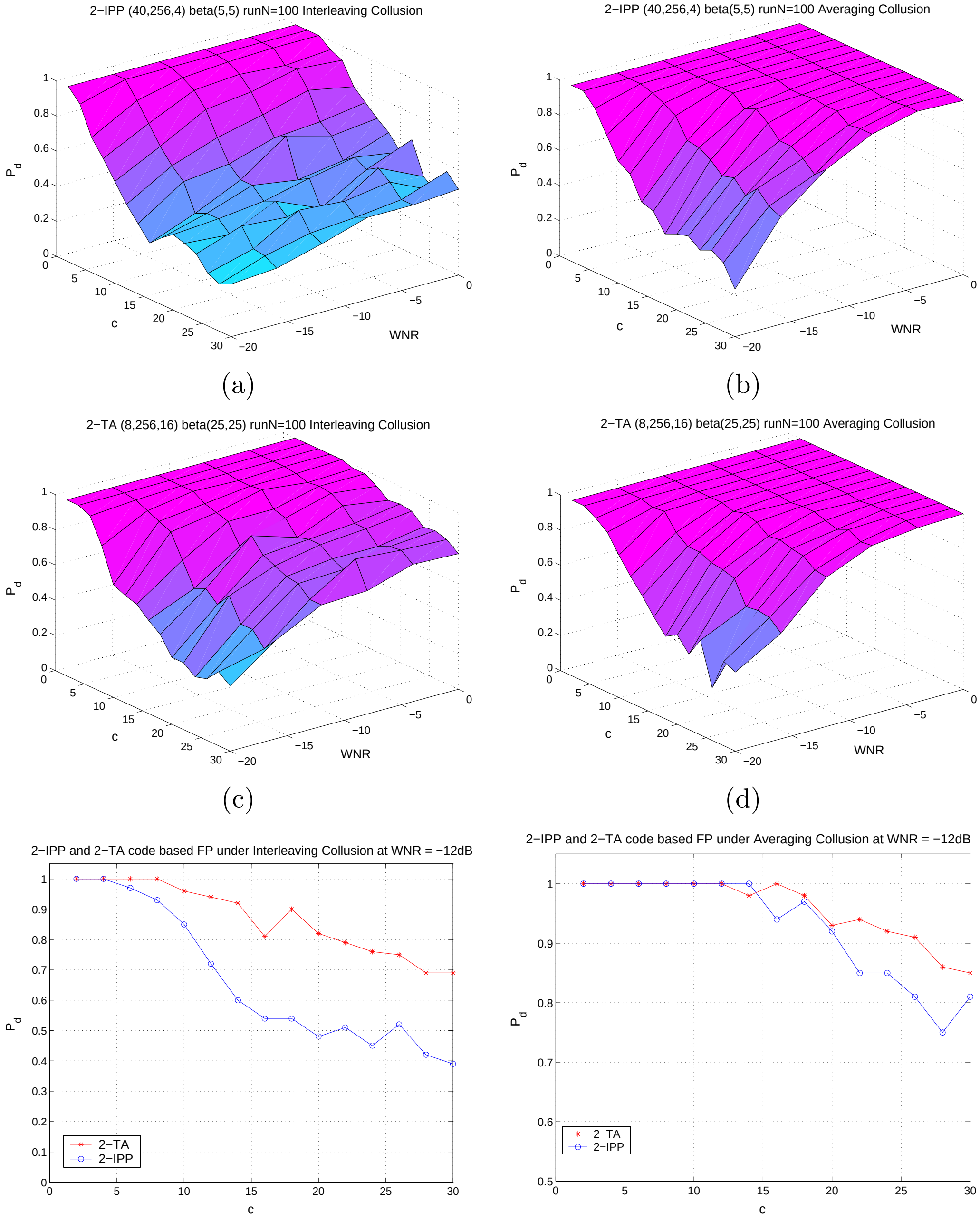
c-IPP Code

A c-IPP code satisfies the condition that any colluded codeword by a coalition of size at most c can be traced back to at least one member of the coalition [56]. A c-TA code is a c-IPP code, but a c-IPP code is not necessarily a c-TA code. Therefore, the set of c-TA codes is a subset of c-IPP codes. In terms of the traceability, the c-TA codes are stronger than those c-IPP codes that are not c-TA codes, which we call proper c-IPP codes. Van Trung et al. propose a method that can be used to construct a proper c-IPP code as follows [65]:

> Let A be an (L_2, N_2, q_2) c-IPP code with code length L_2, codeword number N_2 and alphabet size q_2. Let B be an (L_1, q_2, q_1) c-IPP code with code length L_1, codeword number q_2 and alphabet size q_1. Then the concatenated code C of A and B is an (L_1L_2, N_2, q_1) c-IPP code with code length L_1L_2, codeword number N_2 and alphabet size q_1.

The concatenation of code A and code B is done by replacing each symbol in the alphabet of code A by a codeword in code B. Since a c-TA code is also a c-IPP code, the construction of a proper c-IPP code can be done by concatenating two c-TA codes.

In this section, we are interested in the comparison of c-TA codes with proper c-IPP codes. From this point on, for the sake of brevity we use the term c-IPP codes to refer to proper c-IPP codes.



(a) (b)

(c) (d)

(e) (f)

Figure 3.4: Simulation results for IPP codes and TA codes based fingerprinting systems: the performance of 2-IPP code based system under (a) interleaving collusion and (b) averaging collusion; the performance of 2-TA code based system under (c) interleaving collusion and (d) averaging collusion. The performance of both systems under (e) interleaving collusion and (f) averaging collusion with WNR=−12dB.

3.2.2 Performance Evaluation

c-IPP codes versus c-TA codes

Inequality (3.10) shows the sufficient condition for a code to be a c-TA code, and it does not hold for a c-IPP code. Rewriting inequality (3.10) as

$$\frac{L-D}{L} < \frac{1}{c^2},$$

and combining it with Eqn.(2.18), we can see that a c-TA code has pairwise correlation $\rho_0 < 1/c^2$, while c-IPP code has pairwise correlation $\rho_0 > 1/c^2$. According to the analysis in Section 2.4.3, the fingerprinting system constructed on c-TA code should have better performance than the fingerprinting system employing c-IPP code.

To validate the analysis, we examine the performance of a c-IPP code based fingerprinting system and a c-TA code based fingerprinting system through simulation. For a host signal with length $N = 40,000$, we design two systems that are capable of holding $N_u = 256$ users as follows:

- System 1 is built upon a 2-IPP code (40,256,4) with code length L=40, codeword number $N_u = 256$ and alphabet size q=4. This 2-IPP code is constructed through the concatenation of two 2-TA Reed-Solomon codes (8,256,16) and (5,16,4) following the method proposed in [65]. The pairwise correlation of the fingerprint sequences ρ_0 is 0.3 according to Eqn. (2.18).

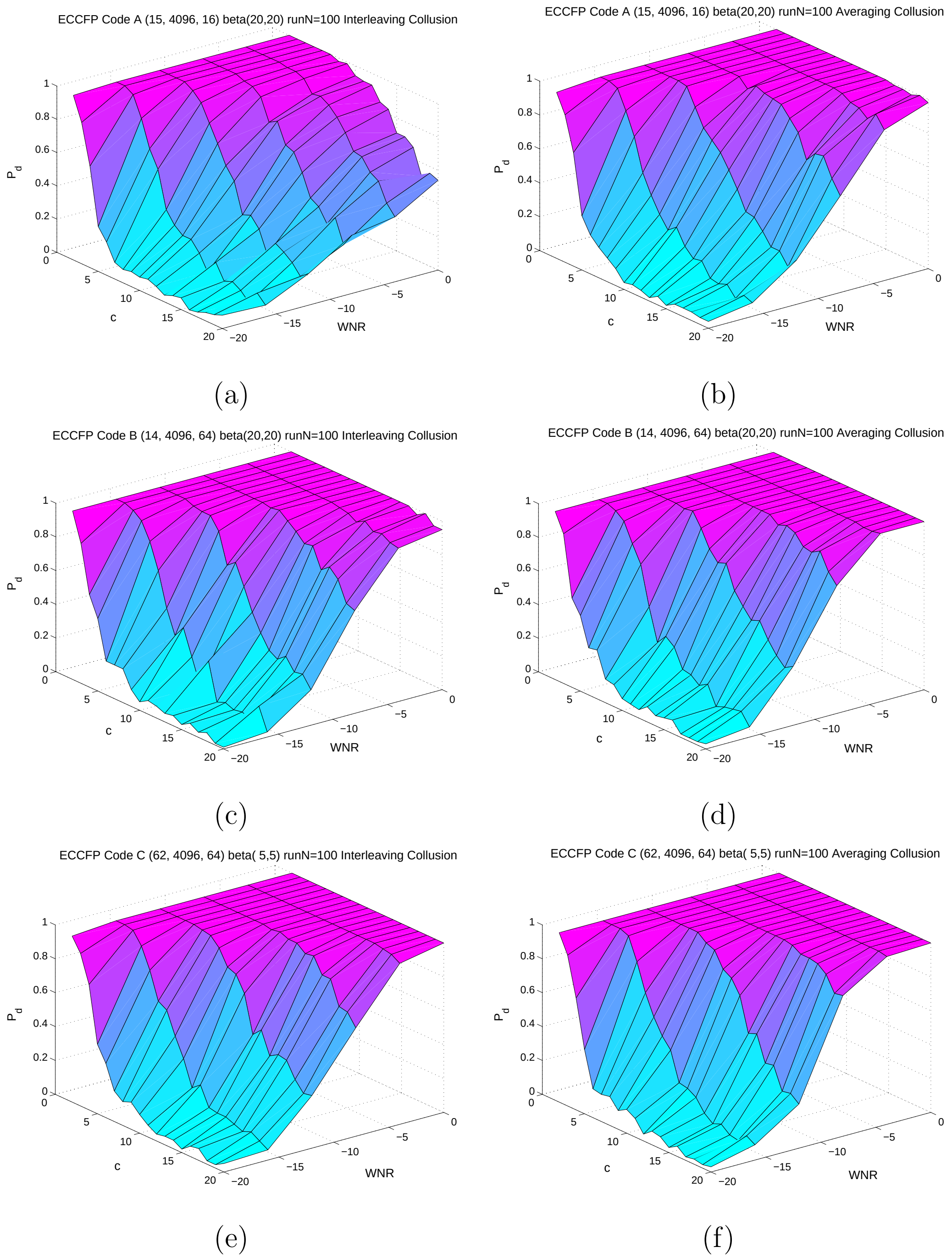
- System 2 is built upon a 2-TA Reed-Solomon code (8,256,16) with code length L=8, codeword number $N_u = 256$ and alphabet size q=16. The pairwise correlation ρ_0 is 0.14.

In both systems, we employ our proposed permuted subsegment embedding technique and choose the same subsegment size 200 for permutation. We examine the probability of catching one colluder P_d of both systems against interleaving collusion and averaging collusion with colluder number c ranging from 2 to 30 and Watermark-to-Noise-Ratio(WNR) ranging from -20dB to 0dB. The simulation results are shown in Fig. 3.4. For ease of comparison, we show the case of WNR$=-12$dB in Fig. 3.4(e) and (f). From the results, we can see that under averaging collusion (Fig. 3.4(b), (d) and (f)) 2-TA code based System 1 has 8% gain in the probability of detection P_d. Under interleaving collusion (Fig. 3.4(a), (c) and (e)), the performance gain can be up to 30%. The results are consistent with our analysis that due to the low pairwise correlation among the fingerprint sequences, 2-TA code based system outperforms 2-IPP code based system in all the cases we examined.

c-TA codes with different parameters

From the above comparison results, we can see that the fingerprint sequences constructed based on a c-TA code have lower correlation than the sequences constructed based on a c-IPP code. This low correlation helps defending against collusion attacks. A TA code is thus preferred in designing the fingerprint sequences. A natural question is that, given a host signal and the number of users the system needs to hold, how should we choose the parameters of TA codes to achieve good collusion resistance.

In the following, we consider TA codes constructed on Reed-Solomon codes over alphabet size of q with dimension t. Examining Eqn. (2.18) we find that in order to get a small ρ_0, we can decrease t and increase L. In order to meet the desired number of users N_u and reduce the dimension t, larger q is preferred. Moreover, for Reed-Solomon code (including extended Reed-Solomon code), $L \leq q + 1$. In



(a)

(b)

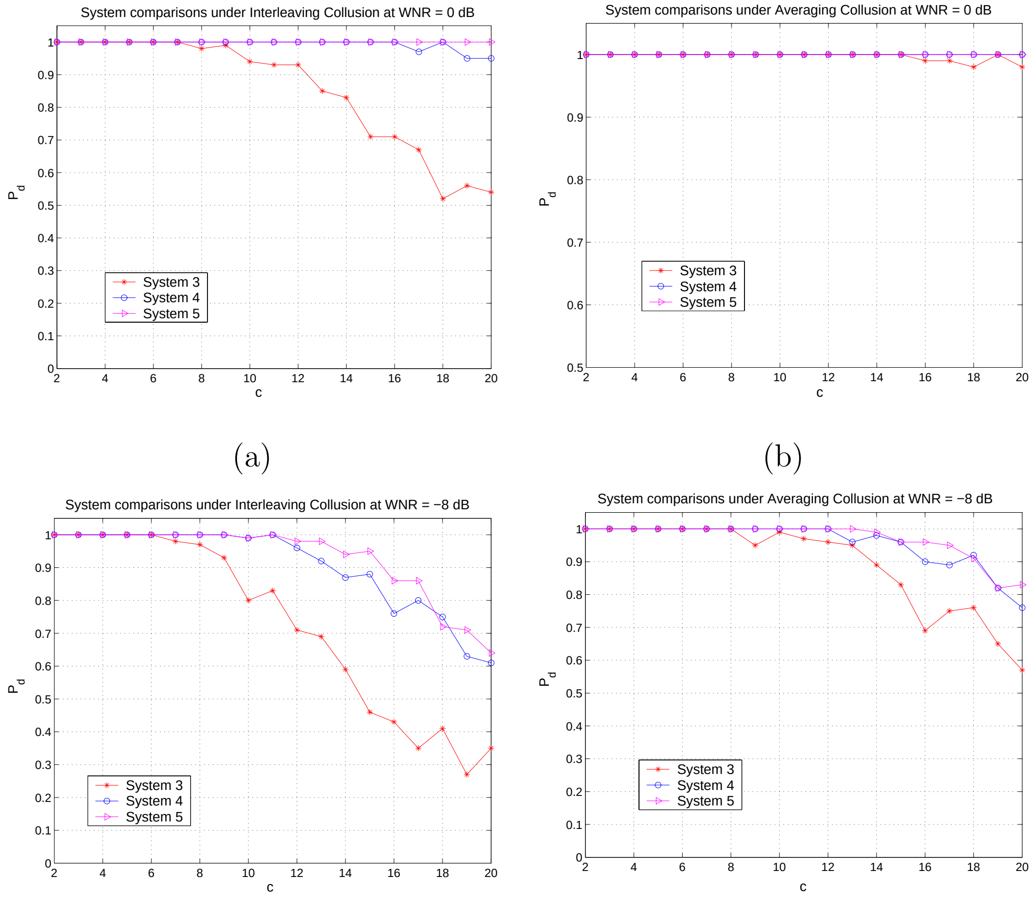
(c)

(d)

(e)

(f)

Figure 3.5: Simulation results for systems with different code parameters under collusion attacks: System 3 under (a) Interleaving Collusion and (b) Averaging Collusion; System 4 under (c) Interleaving Collusion and (d) Averaging Collusion; System 5 under (e) Interleaving Collusion and (f) Averaging Collusion.



(a) (b)

(c) (d)

Figure 3.6: Simulation results for systems with different code parameters under interleaving and averaging collusion at WNR = 0dB and -8dB. (a) Interleaving Collusion with WNR = 0dB; (b) Averaging Collusion with WNR = 0dB; (c) Interleaving Collusion with WNR = -8dB and (d) Averaging Collusion with WNR = -8dB

order to get larger L, a larger q is also preferred. Therefore, our conjecture is that the fingerprinting system constructed on a TA code with a larger alphabet size q and a longer code length L should have better collusion resistance.

To validate our analysis, we examine the collusion resistance of the systems with various parameters through simulations. We construct three fingerprinting systems as follows:

- System 3 is built upon a TA code (15, 4096, 16) with code length $L = 15$, codeword number $N_u = 4096$ and alphabet size $q = 16$. According to Eqn. (2.18), the pairwise correlation ρ_0 is 0.13.
- System 4 is built upon a TA code (14, 4096, 64) with code length L =14, codeword number N_u=4096 and alphabet size q =64. The pairwise correlation ρ_0 is 0.07.
- System 5 is built upon a TA code (62, 4096, 64) with code length L =62, codeword number N_u=4096 and alphabet size q =64. The pairwise correlation ρ_0 is 0.016.

System 3 and System 4 have approximately the same code length but different alphabet size. System 4 and System 5 have the same alphabet size but different code lengths. All the systems are designed to protect a host signal with length $N = 15,000$ and to accommodate $N_u = 4096$ users. We employ the permuted subsegment embedding technique for the fingerprint embedding, and a subsegment size of 50 is chosen for the permutation. We examine the probability of catching one colluder P_d of all three systems against interleaving collusion and averaging collusion, with colluder number c ranging from 2 to 20 and WNR ranging from -20dB to 0dB. We show the simulation results in Fig. 3.5, where the results for WNR = 0dB and -8dB cases are shown separately in Fig. 3.6 for better illustration. Comparing System 3 and System 4, we observe that under averaging collusion (Fig. 3.6(b) and (d)) System 4 with a larger alphabet size has 8% gain in the probability of detection P_d. The performance gain under interleaving collusion (Fig. 3.6(a) and (c)) can be as high as 40%. The comparison of System 3 and System 4 shows that with the same code length and the same subsegment permutation, the system with a larger alphabet size has better performance. Comparing System 4 and System 5, we can see that under both averaging and interleaving collusions, System 5 has about a 5% performance gain due to a longer code length. This small performance gain is because in this particular experimental settings, the pairwise correlations of both System 4 and 5 are very small and close to 0. There is little room for the improvement brought about by the smaller pairwise correlation of System 5. The simulation results of all three systems are consistent with our analysis in Section 2.4.3 in that TA codes with larger alphabet size q and longer code length L result in fingerprint sequences with smaller pairwise correlation, and thus better collusion resistance.

3.2.3 Discussions

The above results show that larger q and L values are preferred in code construction. However, q and L cannot be chosen arbitrarily. There are several constraints on them depending on the code constructions. Specifically, for the Reed-Solomon code construction, we have following constraints:

$$\begin{aligned}
\text{System requirement on the total user number:} \quad q &= \sqrt[t]{N_u}; && (3.11)\\
\text{Reed-Solomon code construction constraint:} \quad L &\leq q+1; && (3.12)\\
\text{Orthogonality of the FP sequences for each segment:} \quad q &\leq \frac{N}{L}. && (3.13)
\end{aligned}$$

where N is the host signal length, N_u is the total number of users, q is the alphabet size and L is the code length. Taking L as the maximum value $q+1$, we get from (3.13) that

$$q(q+1) \leq N; \tag{3.14}$$

which means the upper bound of q value is roughly on the order of $\sqrt{N}$. Usually, in multimedia fingerprinting the host signal length $N >> N_u$ and $t \geq 2$ for Reed-Solomon codes. Therefore, Eqn. (3.11) is a more stringent requirement on q. In Eqn. (3.11), the dimension t can be used to achieve the desired trade-off between the collusion resistance and the computational complexity in detection which is $O(qN)$ according to Section 3.1.4. For example, in applications where the detection computation resources are very limited, we can first choose a small q to achieve a low complexity detection and then adjust t value to reach a large user capacity. On the other hand, if the detection complexity is not a major concern, the system designer can fix t to be 2 and $q = \sqrt{N_u}$ to minimize the correlation for high collusion resistance. We can see that these parameters provide a tradeoff between collusion resistance and efficiency, and they should be chosen according to the priority of various design requirements. Notice that the extreme case of $t = 1$ reduces to orthogonal fingerprinting which has better collusion resistance but high computational complexity in detection [32].

Other c-TA code constructions can be analyzed in a similar way. It is worth mentioning that the TA code proposed in [66] can be regarded as a TA code with dimension t lying between 1 and 2, which offers a fine adjustment on the trade-off between the collusion resistance and detection efficiency.

3.3 Chapter Summary

In this chapter, we focus on improving the collusion resistance of the ECC based fingerprinting while retaining its advantages in detection complexity and fast distribution. We have discovered a gap in the collusion resistance of ECC based fingerprinting between the averaging and interleaving collusions. Our analysis on the gap suggests a great need of jointly considering the coding, embedding, and detection issues, and inspires to the proposed technique of permuted subsegment embedding. Experimental results demonstrate that the proposed technique can substantially improve the collusion resistance of ECC based fingerprinting, while inheriting the advantage in detection complexity and efficient distribution.

Based on the proposed technique, we then examine the collusion resistance of the coded fingerprinting. The results show that for a given host signal the pairwise correlation among fingerprint sequences is a key indicator of the collusion resistance, the lower the correlation the higher the collusion resistance. According to this principle, c-TA codes can be used to introduce a lower correlation among fingerprint sequences and thus is preferred over c-IPP codes in fingerprint design. Furthermore, a TA code with a larger alphabet size and a longer code length can provide better collusion resistance. The fingerprinting code construction provides a systematic way to introduce the correlation and to achieve a desired trade-off between the collusion resistance and detection efficiency.

Chapter 4

GRACE: Group Based Joint Coding and Embedding

Our second joint coding and embedding 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 non-coded orthogonal fingerprinting [70]. 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/she is in. The group information is used in the detection to narrow down the suspicious user set. Such kind of prior knowledge on 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 the Chapter 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. 2.4(a) 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 [31]. 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.

4.1 Incorporating Grouping in Coded Fingerprinting

4.1.1 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.
```

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 in Chapter 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 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/her user subcode. This process is illustrated in Fig. 4.1 and 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)* [66].

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

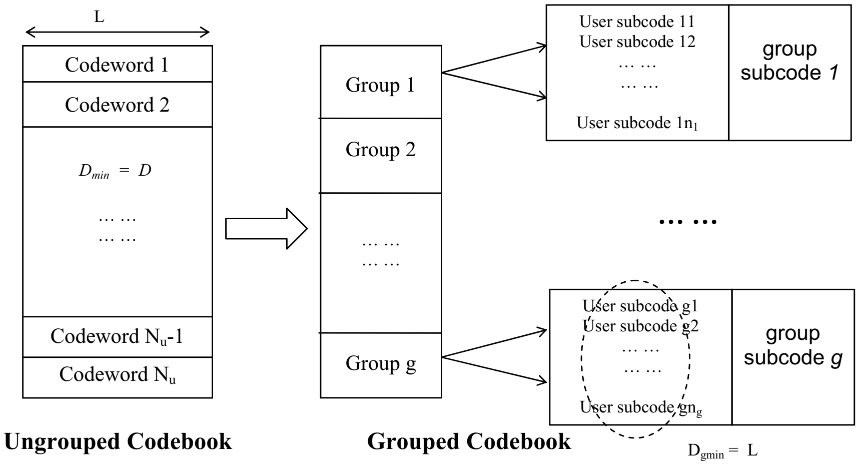


Figure 4.1: Fingerprint codeword construction for GRACE fingerprinting.

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.

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 [73]. More specifically, we use the sequences $\{\mathbf{u}_j, j = 1, ..., q\}$ 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 $\{\mathbf{a}_i, i = 1, ..., g\}$ 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 k^{th} segment of user j who belongs to group i as

$$\mathbf{s}_{ijk} = \sqrt{1-\rho}\mathbf{u}_{sym(j,k)} + \sqrt{\rho}\mathbf{a}_i, \tag{4.1}$$

where the function $sym(j,k)$ is used to retrieve the symbol for the k^{th} segment from the j^{th} 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 k^{th} 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 trade-off 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 happens 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 of each segment $N'_s = N/(L + L_g)$ is.

4.1.2 Fingerprint Detection

At the detector side, the embedded group information can be used to facilitate the detection by a two-level detection scheme. First, 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 2.3 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 non-blind correlation detector. The detection statistic with respect to group i is

$$T_G(i) = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{b}_i}{||\mathbf{b}||}, \quad i = 1, 2, ..., g, \tag{4.2}$$

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, $\mathbf{b}_i^T = [\mathbf{a}_i^T ... \mathbf{a}_i^T]$ 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. In this work, we choose the threshold to be adaptive, i.e. $h = \gamma \max_{k=1,...,g} T_G(k)$, where γ is a parameter to adjust the threshold. The details of the adaptive detection will be presented in Section 4.3. 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 Eqn.(2.6) to correlate 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 non-blind correlation detector. The decoding to a specific colluder is then conducted on the segments for user subcode within the extracted suspicious groups.

4.1.3 Experimental Results

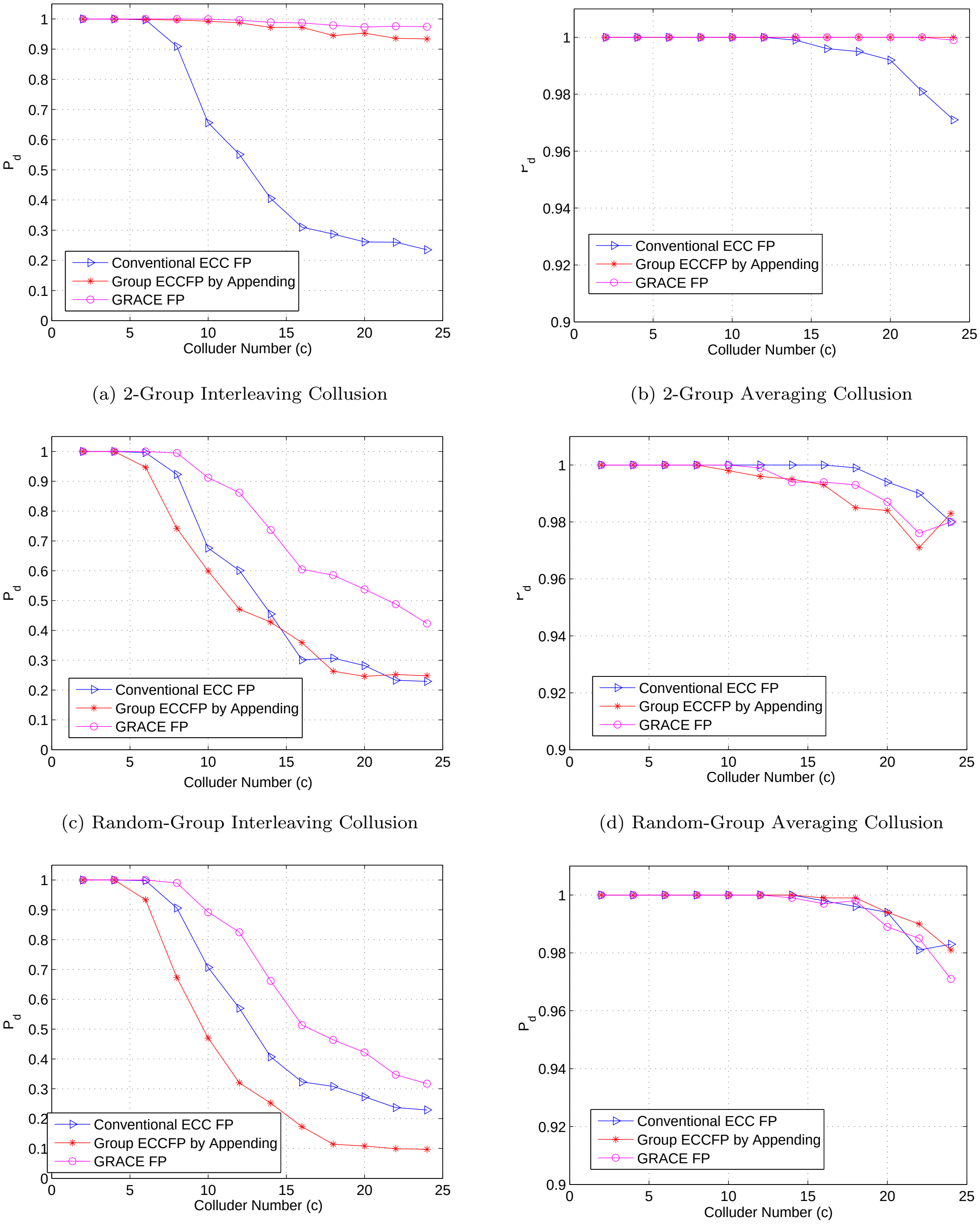
In this section, we demonstrate the effectiveness of the proposed GRACE fingerprinting through experiments. To build the user subcode, we employ a Reed-Solomon code with $q = 32, L = 30, N_u = 1024$, $D = 29$, and rearrange it into 32 groups using the algorithm described in Section 4.1.1. Inside each group, there are 32 codewords mutually orthogonal to each other. We choose $\rho = 1/7$ in Eqn.(4.1) 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 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 4.1.1 as the group subcode, and construct *i.i.d.* Gaussian signals with 3×10^4 signal samples to emulate the host signal.

Interleaving collusion 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 WNR = 0dB 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 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 results of P_d under interleaving collusion and averaging collusion are shown in Fig. 4.2(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 6 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. 4.2(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 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 the colluders come from distinct groups (i.e. the num-



(a) 2-Group Interleaving Collusion

(b) 2-Group Averaging Collusion

(c) Random-Group Interleaving Collusion

(d) Random-Group Averaging Collusion

(e) Distinct-Group Interleaving Collusion

(f) Distinct-Group Averaging Collusion

Figure 4.2: Performance comparison of the proposed GRACE fingerprinting, group ECC fingerprinting by appending and the conventional ECC fingerprinting in terms of probability of detection P_d versus the colluder number c at WNR = 0dB.

ber of groups equals the number of colluders c). The results under interleaving and averaging collusion are shown in Fig. 4.2(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 happens 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 non-group 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 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 WNR = 0dB presented in Fig. 4.2, we also examined the cases of low WNRs, and the comparative results are similar to 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.

4.1.4 Discussions

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 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/her share only to the group subcode positions and other colluders from a different group only contribute to user subcode positions. We call this *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 the user subcode, he/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 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/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.

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 2.4.1, 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.

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 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 [70] 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.

4.2 Combining GRACE with Permuted Subsegment Embedding

Earlier in Chapter 3, we have 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. 4.3. 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,

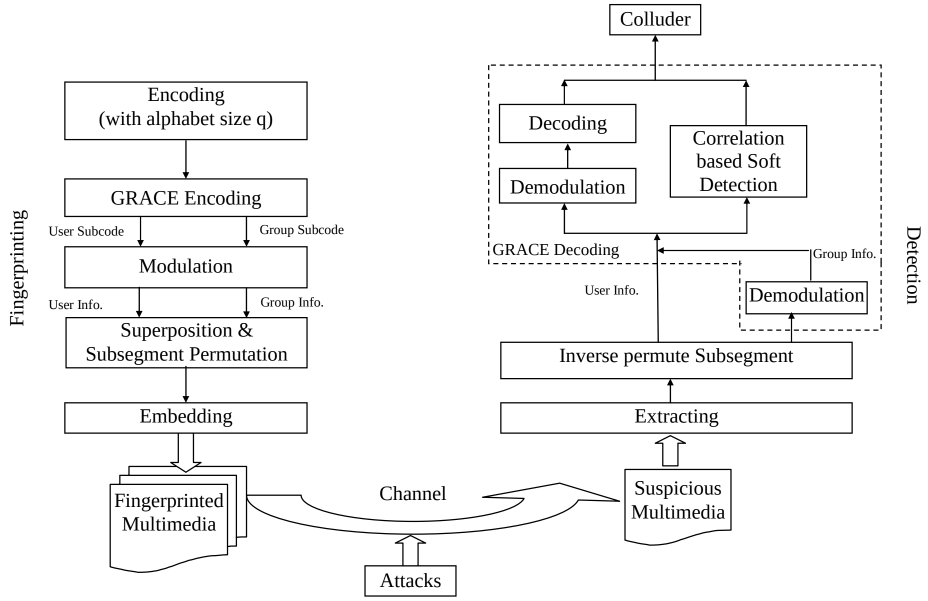
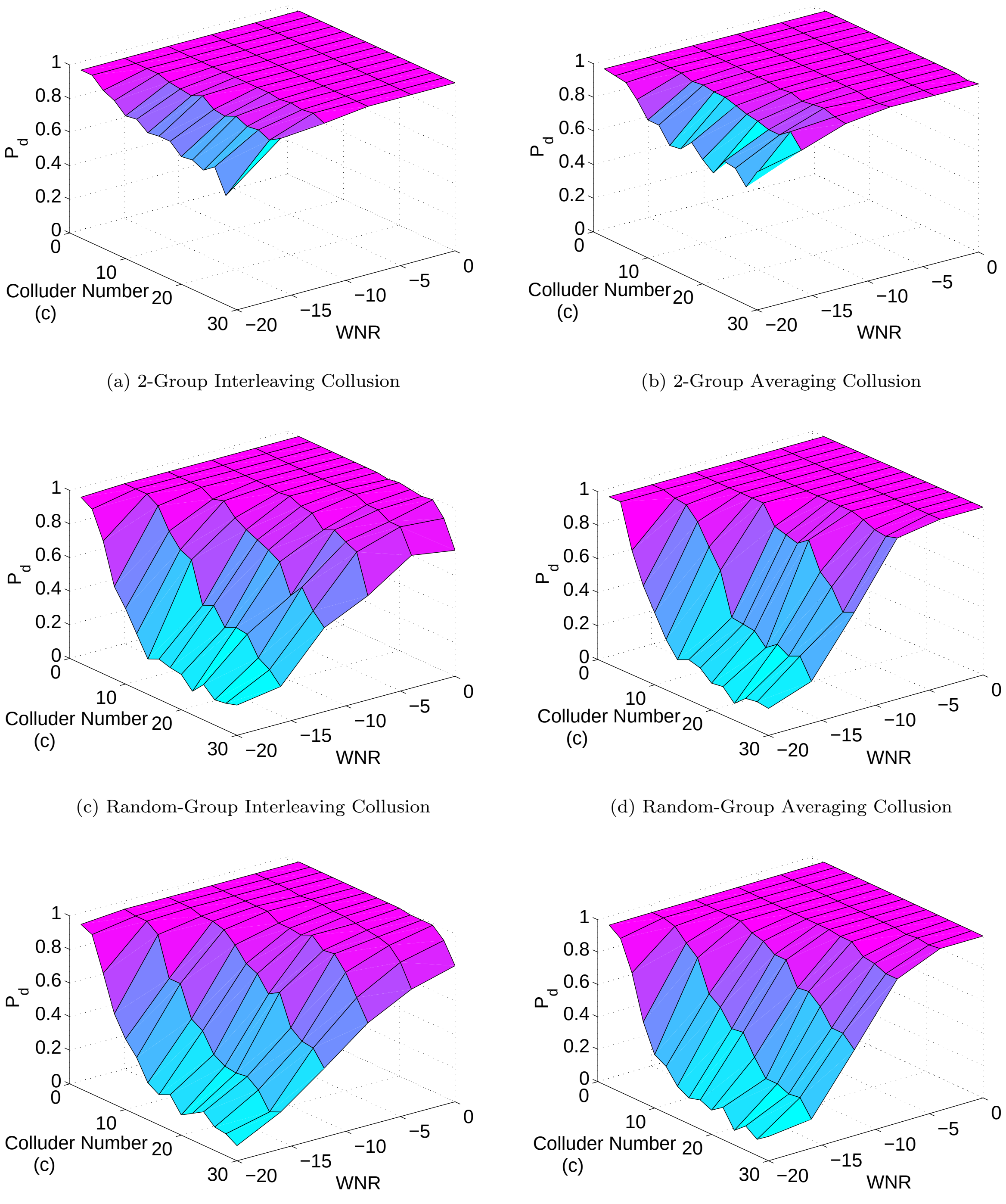


Figure 4.3: Proposed framework of coded multimedia fingerprinting combining GRACE with permuted subsegment embedding.

namely, the extraction of the group information followed by the soft detection of the colluder using Eqn.(2.6) within the extracted groups. We demonstrate the performance of the combined fingerprinting system through simulations on the same system as we have examined in Section 4.1.

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. 4.4(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. 4.4(a)-(b)), the grouping strategy boosts the detection probability



(a) 2-Group Interleaving Collusion

(b) 2-Group Averaging Collusion

(c) Random-Group Interleaving Collusion

(d) Random-Group Averaging Collusion

(e) Distinct-Group Interleaving Collusion

(f) Distinct-Group Averaging Collusion

Figure 4.4: Performance of the proposed GRACE fingerprinting with permuted subsegment embedding technique: Probability of detection P_d vs. the colluder number c and WNR.

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 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 a test signal.

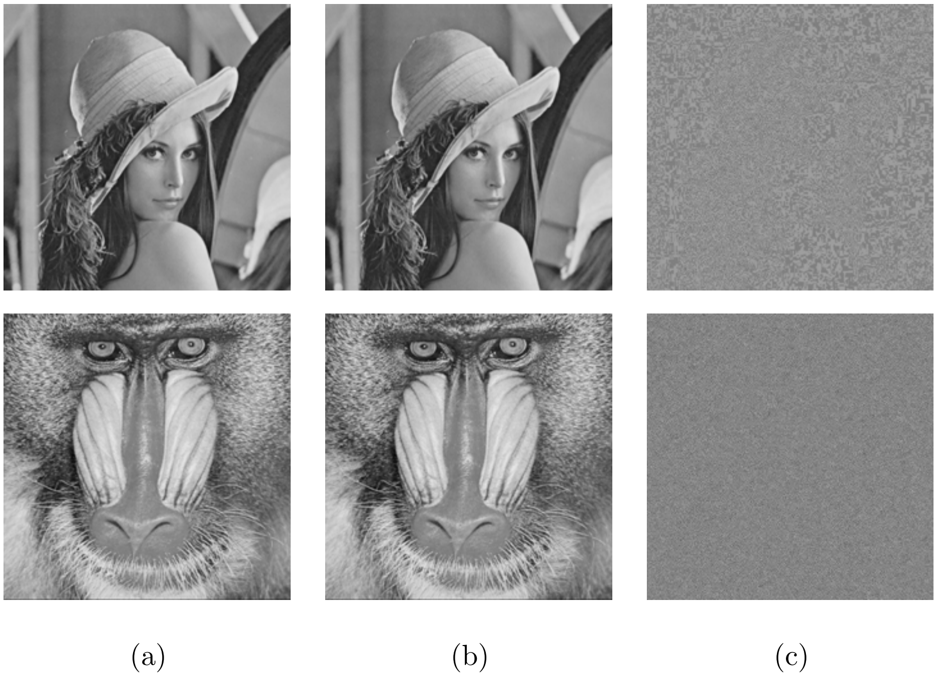
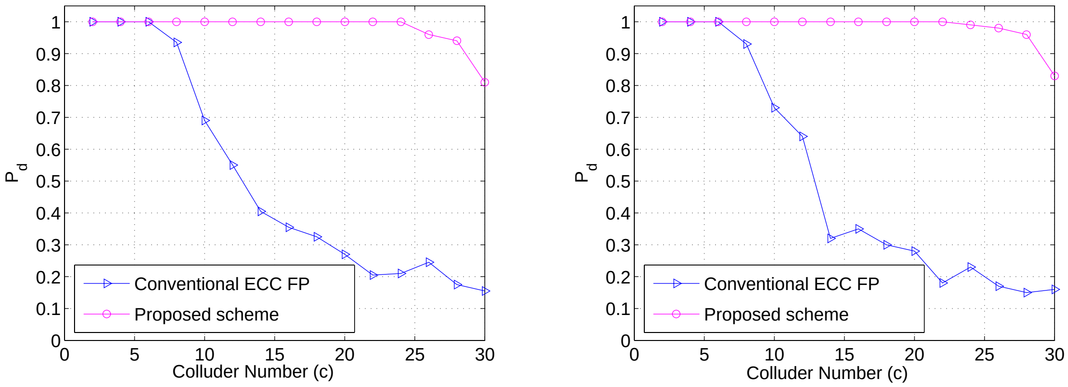


Figure 4.5: (a) Original images; (b) Fingerprinted images; (c) Corresponding difference images (amplified by a factor of 10).

We select 512×512 Lena and Baboon as original images to demonstrate the



(a) (b)

Figure 4.6: Experimental results on real images of (a) Lena and (b) Baboon under interleaving collusion.

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. We employ the same coding setup as in Section 2.4.3 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 41.6dB for Lena and 33.2dB for Baboon. Fig. 4.5 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. 4.6, 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 detection probability as high as 0.98. We also examined the averaging collusion scenario, and the improvements for both cases are consistent with earlier results on synthetic signals.

4.3 Adaptive Detection for Group-based Fingerprinting

Results in both our GRACE fingerprinting and the prior work by Wang et al. [70] have shown that the grouping strategy increases the collusion resistance of both non-coded and coded fingerprinting schemes when the grouping reflects the collusion pattern. However, when the grouping does not match the collusion pattern well, the group information in the colluded signal would be significantly reduced and group detection may become unreliable, which would lead to a lower detection accuracy than the non-grouped schemes. In this section, we explore how to design and adapt the system to combat a wide range of collusion patterns, especially when the actual collusion pattern is considerably different from the grouping patterns in the system design [38]. For easy analysis, we choose to examine the adaptive detection on non-coded group fingerprinting by Wang et al. [70] and the results on GRACE fingerprinting will be also reported.

4.3.1 Non-Coded Group-based Fingerprinting

The group-based fingerprint construction in [70] takes advantage of the prior knowledge on the collusion pattern by putting users who are likely to collude into the same group and introducing correlation among their fingerprints. The constructed fingerprint consists of user information and group information. The fingerprint sequence for user j who belongs to group i is constructed as

$$\mathbf{s}_{ij} = \sqrt{1-\rho}\mathbf{e}_{ij} + \sqrt{\rho}\mathbf{a}_i, \tag{4.3}$$

where ρ is used to adjust the correlation among the users' fingerprints inside one group as well as the ratio of the fingerprint energy assigned to group information; $\mathbf{e}_{ij}$'s are the spreading sequences for individual user information and they are mutually orthogonal to each other; $\mathbf{a}_i$'s are the spreading sequences for group information and they are orthogonal to each other and also to the user spreading sequences $\mathbf{e}_{ij}$'s.

The collusion attack considered in this section is K-user averaging collusion plus additive noise. A number of other collusions based on order statistics, such as minimum collusion attack, have been shown to be well approximated by such a model [82]. The colluded version $\mathbf{z}$ follows:

$$\mathbf{z} = \frac{1}{K}\sum_{i=1}^{L}\sum_{j\in S_{ci}} \mathbf{y}_{ij} + \mathbf{d} = \frac{1}{K}\sum_{i=1}^{L}\sum_{j\in S_{ci}} \mathbf{s}_{ij} + \mathbf{x} + \mathbf{d}, \tag{4.4}$$

where $\mathbf{x}$ is the host signal and L is the total number of groups. $S_{ci} \subseteq \{1, ..., M\}$ is a subset with size $|S_{ci}| = k_i$ and contains the members of group i who participate in the collusion attack. M is the number of users inside each group. The additional distortion is modelled as an *i.i.d.* additive Gaussian noise $\mathbf{d}$ with zero-mean and variance σ_d^2.

The detection scheme consists of two stages [70]. The group information is first used to detect the guilty groups and then the colluder detection is conducted within the guilty groups. A correlation based method is employed for group detection [70], that is,

$$T_G(i) = \frac{(\mathbf{z}-\mathbf{x})^T(\mathbf{s}_{i1} + \mathbf{s}_{i2} + ... + \mathbf{s}_{iM})}{\sqrt{\|\mathbf{s}\|^2[M + (M^2 - M)\rho]}}, \tag{4.5}$$

whose probability density function (pdf) is

$$p(T_G(i)|K,k_i,\sigma_d^2)=\begin{cases}N(0,\sigma_d^2), \quad \text{if} \quad k_i=0,\\ N(\frac{k_i\|\mathbf{s}\|\sqrt{1+(M-1)\rho}}{K\sqrt{M}},\sigma_d^2), \quad \text{o.w.}\end{cases} \tag{4.6}$$

where $\|\mathbf{s}\|$ is the strength of the fingerprint and it is the same for each user due to the equal-energy fingerprint construction. The i-th group is declared as guilty if $T_G(i)>h_G$, where h_G is the threshold for group detection.

The colluder detection is performed within the guilty groups through a correlation based detector:

$$\hat{j}_i=arg_{j=1}^{M}\{T_{ei}(j)\geq h\}, \tag{4.7}$$

where h is a threshold to determine the colluders and the detection statistic for user information is

$$T_{ei}(j)=\frac{\sqrt{1-\rho}(\mathbf{z}-\mathbf{x})^T\mathbf{e}_{ij}}{\sqrt{\|\mathbf{s}\|^2}}.$$

The group threshold h_G and the user threshold h are chosen such that the false alarm on group detection is bounded by α_1 and the probability for an innocent user to be declared as colluder is bounded by α_2:

$$\alpha_1\triangleq Pr(T_G(i)\geq h_G|k_i=0)=Q(\frac{h_G}{\sigma_d}); \tag{4.8}$$

$$\alpha_2\triangleq Pr(T_{ei}(j)\geq h|j\notin S_{ci})=Q(\frac{h}{\sigma_d\sqrt{1-\rho}}). \tag{4.9}$$

Thus, $h=Q^{-1}(\alpha_2)\sigma_d\sqrt{1-\rho}$ and $h_G=Q^{-1}(\alpha_1)\sigma_d$, where the Q-function is defined as $Q(t)=\int_t^\infty 1/\sqrt{2\pi}\exp(-x^2/2)dx$. The overall probability of catching one colluder P_d and the probability of false alarm P_{fp} can be calculated as [70]:

$$\begin{aligned}P_d &= \sum_{i=1}^{l}q_i\prod_{j=1}^{i-1}(1-q_j),\\ P_{fp} &= [1-(1-p_{l+1})^{L-l}]+(1-p_{l+1})^{L-1}\sum_{i=1}^{l}p_i\prod_{j=1}^{i-1}(1-p_j),\end{aligned} \tag{4.10}$$

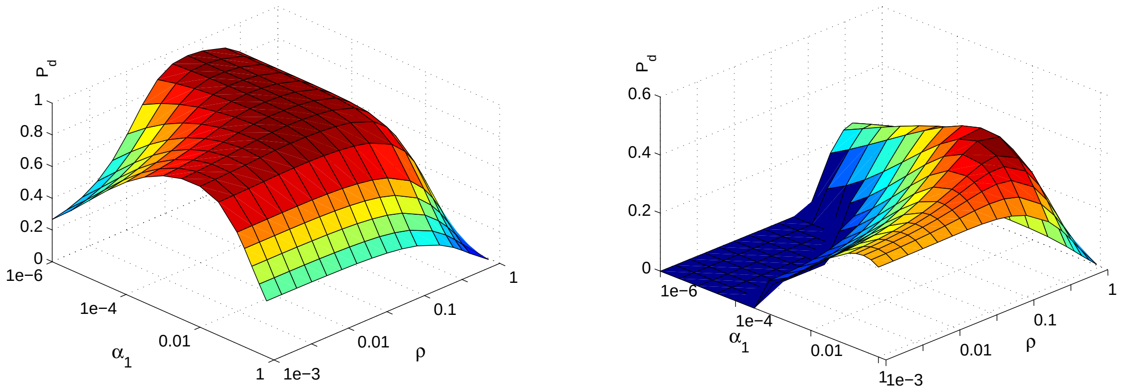
where l is the number of guilty groups, p_i is the probability of a false alarm event in which at least one innocent user from group i is declared as colluder, and q_i is the probability of a correct detection event in which at least one true colluder from group i is declared as guilty. Both p_i and q_i contain multiple terms with the form of a Q-function, and they are functions of the three parameters ρ, α_1 and α_2, which should be chosen in such a way that

$$\begin{aligned}\{\alpha_1, \alpha_2, \rho\} = \arg \max_{\alpha_1, \alpha_2, \rho} P_d(\alpha_1, \alpha_2, \rho) \\ \text{subject to} \quad P_{fp}(\alpha_1, \alpha_2, \rho) \leq \epsilon.\end{aligned} \tag{4.11}$$

4.3.2 Performance Evaluation of Group-based Fingerprinting

Due to multiple terms of Q-function in Eqn. (4.10), closed-form relations between systems performance P_d and system parameters ρ and α_1 on an arbitrary collusion pattern cannot be found. Commonly used analytical bounds are usually not very tight for our performance evaluation purpose. We use the system in [70] as an example and examine the optimization problem of (4.11) through numerical evaluation to approximate the optimal solution to the parameter settings.

In the example system, we choose fingerprint sequence with length 10^4 and examine the detection performance at Watermark-to-Noise-Ratio (WNR) of 0dB. There are totally $L = 100$ groups and $M = 60$ users in each group; a total of $K = 64$ users participate in averaging collusion. These values are chosen based on the guidelines from the orthogonal fingerprinting study in [71] in order to maintain good collusion resistance of mutually orthogonal group-fingerprint on the group detection level as well as the orthogonal user fingerprint within each group. We consider the worst-case scenario in detecting at least one of the true colluders, where all the colluders are evenly distributed among l guilty groups, i.e. $k_i = K/l$. The colluder detection performance at $P_{fa} = 0.1$ under two different collusion patterns of $l = 4$ and $l = 16$, respectively, is shown in Fig. 4.7. We can see that the values of ρ and α_1 significantly affect the detection probability P_d, and the values of these two parameters that achieve the optimal P_d under different collusion patterns are different. Next, we take a close look at the effect of these two parameters.



(a) (b)

Figure 4.7: P_d versus ρ and α_1 under averaging collusion attack from (a) 4 groups and (b) 16 groups.

The effect of ρ From the fingerprint construction of Eqn. (4.3) and group detection analysis in Eqn. (4.6), we can see that the larger the ρ is, the higher the correlation is and the better the group detection is. But large ρ also reduces the energy that can be allocated to represent individual user and therefore may reduce the detection accuracy.

We take the same system as that of Fig. 4.7 and numerically examine the effect of different ρ's on the collusion resistance. Three scenarios with $l = 4, 8, 16$ are studied, and the results of P_d at $P_{fa} = 0.1$ are shown in Fig. 4.8, along with the results of orthogonal fingerprinting [71]. From Fig. 4.8, we can see that the difference of P_d's under different ρ values can be up to 70%, and there is no ρ value that reaches the best performance under all the collusion patterns. We also note that ρ must be determined during the design of the system without the knowledge of the actual collusion patterns. Thus, ρ value should be chosen to achieve a good performance trade-off under a broad range of anticipated collusion patterns. For the experiment settings in Fig. 4.8, relatively small ρ around 0.1, instead of a higher setting of 0.4 as in [70], leads to a better trade-off among various collusion patterns. Therefore, we choose $\rho = 0.1$ in the following study.

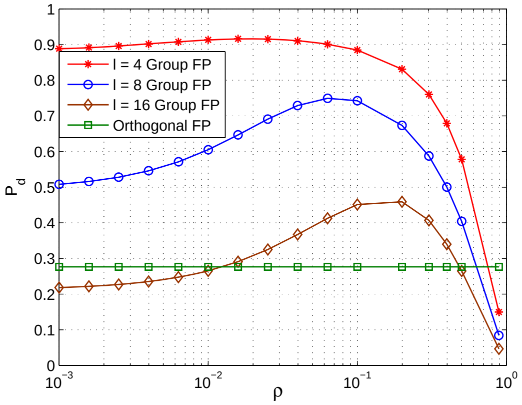
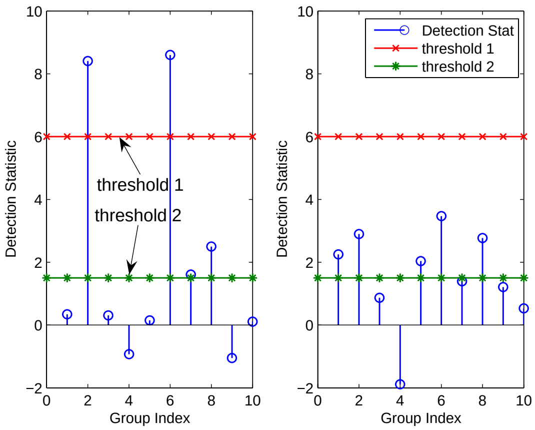


Figure 4.8: P_d vs. ρ under averaging-collusion attack.

The intuition of choosing relatively small ρ is as follows. When the grouping strategy captures the collusion pattern, many colluders come from the same group i, and thus they have the same group information $\sqrt{\rho}\mathbf{a}_i$ in their fingerprints. Under averaging collusion, the energy of this group information will not be reduced much, as can be seen from the mean value of detection statistic $T_G(i)$ for guilty group in Eqn. (4.6). In the term $\frac{k_i\|s\|\sqrt{1+(M-1)\rho}}{K\sqrt{M}}$, although the total number of colluders K can be quite large, k_i, the number of colluders inside group i, is also large. As a result, even with a small ρ, the mean value of $T_G(i)$ would be reasonably high to distinguish the guilty groups from innocent groups, leading to a high detection probability. When the grouping strategy does not reflect the collusion pattern, k_i would be quite small for all the groups. Thus $T_G(i)$ for guilty groups would have a small mean value, leading to a less reliable group detection. To achieve a high colluder detection accuracy, we would then rely more on the user information part of the fingerprint in the second step detection. Relatively small ρ value keeps the energy of user information strong to distinguish each individual user and help achieve a high probability of detection. Therefore, relatively small ρ provides a good performance trade-off under a wide range of collusion patterns.



(a) (b)

Figure 4.9: Group detection statistics for colluders from (a) 2 groups, and (b) 4 groups.

The Effect of α_1 and h_G In the two-step detection strategy, the false alarm of group detection α_1 and group detection threshold h_G are related through the false alarm probability of group detection in Eqn. (4.8), and they determine the accuracy of group detection. If this step is accurate enough, as illustrated by "threshold 1" in Fig. 4.9(a) and "threshold 2" in Fig. 4.9(b), many innocent

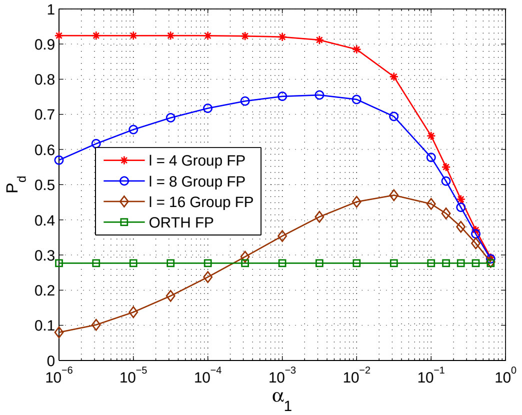


Figure 4.10: P_d vs. α_1 under averaging-collusion attack.

users are successfully filtered out in the first step, so the second step of locating individual colluders only needs to focus on a small set of users and can achieve a high detection accuracy. However, if the group detection is not accurate, the second step of detecting colluders cannot take advantage of the grouping strategy and the system performance may become worse than the non-grouping case. There are two cases of inaccurate group detection. One is when the threshold is too low that many innocent groups are passed onto the second step detection, as shown in Fig. 4.9(a) with “threshold 2”. The increase in the number of innocent users incurs higher probability of false alarm during the colluder detection, leading to a low overall detection accuracy. The other case is when the threshold is so high that many groups, including true guilty groups, fail to pass the group detection. This is shown in Fig. 4.9(b) for “threshold 1”. With few true colluders being inside the suspicious user set, it is difficult for the second step detection to correctly identify the colluder. Therefore, α_1 should be chosen carefully in the detection.

We examine different α_1 values for the above system settings with $\rho = 0.1$ according to the above discussions, and show the results of P_d for a fixed $P_{fa} = 0.1$ in Fig. 4.10. We see that when colluders come from only 4 groups, a low value of $\alpha_1 = 10^{-6}$ provides the highest detection probability P_d, and a wide range of α_1 values can lead to near optimal P_d. When colluders spread to more and more groups, the value of α_1 achieving the highest P_d increases to 3×10^{-3} for 8-group collusion and to 3×10^{-2} for 16-group collusion. P_d for low α_1 value at around 10^{-6} significantly drops and becomes even lower than that of orthogonal fingerprinting. Collectively considering various collusion patterns, we can see that for this particular system that we have examined, a less stringent setting for α_1, such as 10^{-2}, has a better trade-off in the detection performance. This is because when most colluders come from only a few groups as shown in Fig. 4.9(a), with high probability the group detection statistics for guilty groups have large values and are much higher than that of innocent groups. In this case, employing a moderate threshold rather than a low one can help filter out more innocent groups and pass most of the guilty groups to the second detection step, leading to a high overall detection probability as shown in Fig. 4.10 for $l = 4, 8$. When colluders come from multiple groups, the group detection statistics of guilty groups are reduced and not very distinguishable from those of innocent groups as shown in Fig. 4.9(b). Compared with a high threshold, a moderate threshold enables the detector to include more guilty groups into the second step detection to achieve a better performance as shown in Fig. 4.10 with $l = 16$.

An important finding suggested by the above analysis is that a variable threshold in group detection is necessary to deal with different collusion patterns. In fact, the detector has the freedom to adjust α_1 according to its findings and to adapt the threshold value for group detection according to the detection statistics. In the next section, we present an adaptive group detector and examine its performance.

4.3.3 Adaptive Detection

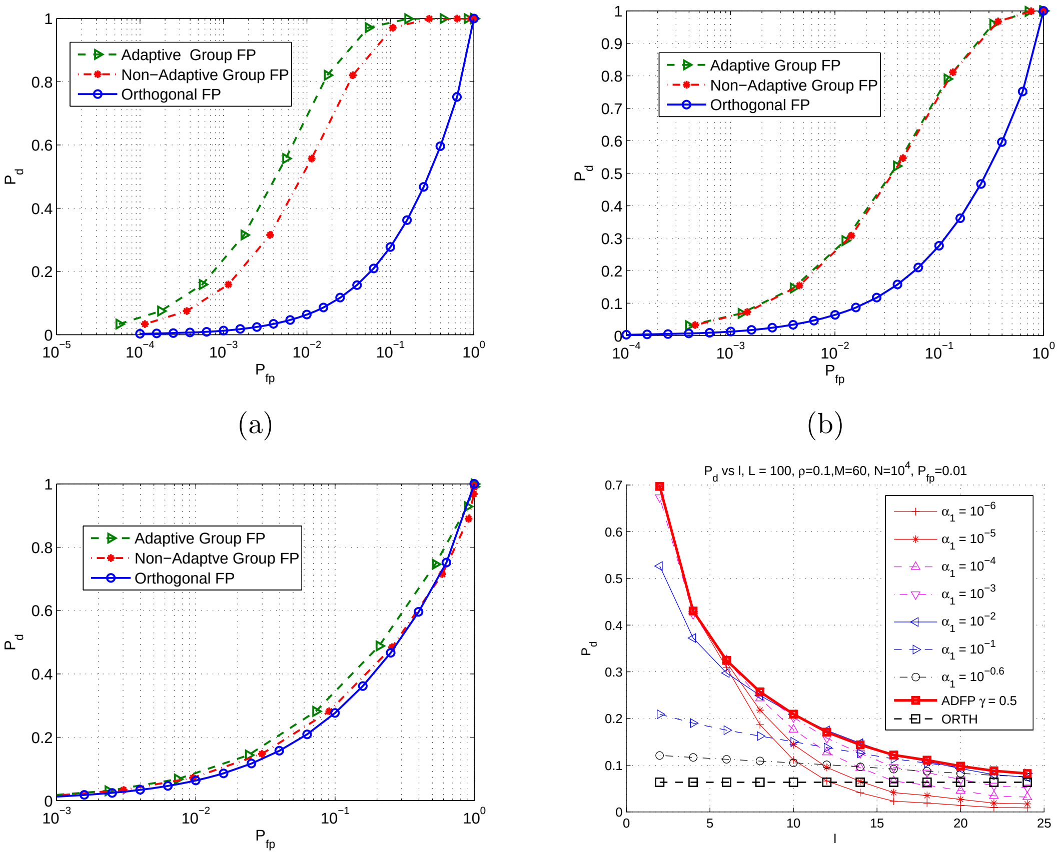
Studies in Section 4.3.2 suggest that an ideal group detector should be able to adjust the threshold for various collusion patterns. To achieve this, the detector first needs to collect information to infer the underlying collusion pattern. From the group detection statistics, we can see that when colluders come from a small number of groups as system designers expect, the detection statistics are high for guilty groups and low for innocent groups. This is because the shared group information in colluders' fingerprints is preserved after collusion and has relatively high energy. On the other hand, when collusion pattern does not match the grouping, the detection statistics for guilty groups are low and not very distinguishable from innocent groups. Another scenario is that a non-trivial part of the colluders come from one group i, and the other colluders scatter over several groups. The detection statistic for group i will be high with high probability, although the detection statistics for others groups are low.

As the detection statistics on the group information reflect the collusion pattern, we examine how to exploit the detection statistics to facilitate the group detection. A good and easy-to-obtain indicator for collusion patterns is the maximal group detection statistic T_{Gmax}, which can be used by the adaptive detector as a baseline to set threshold. For example, the threshold h_G can be made to be proportional to T_{Gmax}, i.e. $h_G = \gamma T_{Gmax}$ with $0 < \gamma < 1$, so that group i is declared guilty if $T_G(i) > \gamma T_{Gmax}$. We set γ at 0.5 in our experiments. The probability of group i being caught, $P_{Gd}(i)$, can be obtained by

$$\begin{aligned} P_{Gd}(i) &= Pr(T_G(i) > \gamma T_{Gmax}) \\ &= \int_{-\infty}^{\infty} \prod_{r \neq i} Pr(T_G(r) < t/\gamma) f_{T_G(i)}(t) dt, \end{aligned} \qquad (4.12)$$

where $T_G(i)$ follows a Gaussian distribution according to Eqn. (4.6). We can then analyze the colluder detection probability P_d and false alarm probability P_{fp} by following similar strategies as in [70]. By choosing the threshold in this way, at least one group will always be identified as possibly containing traitors and passed onto the second stage in detection to pinpoint the colluders, whereas in the non-adaptive detection it is possible that all groups are vindicated in the first stage and never make it to the second stage of detection even when there are traitors.

We examine the performance of the proposed adaptive group detection and choose the same system as studied in the previous sections, i.e. $L = 100, M = 60$, and $K = 64$. We consider three different collusion scenarios of $l = 24, 8, 2$, and compare the results with the corresponding non-adaptive detection cases. For comparison, we choose $\alpha_1 = 10^{-2}$ for the non-adaptive scheme, as this setting provides the best performance for a non-adaptive detector according to the studies in Section 4.3.2. The results in Fig. 4.11(a) and (b) show that when all the colluders come from only a few groups so that the group structure well reflects the collusion pattern, the adaptive detection can perform better than the non-adaptive scheme; and both schemes outperform the orthogonal fingerprinting that does not exploit the group structure. The improvement of adaptive detection can be up to 10% in P_d for the same P_{fp} over non-adaptive scheme, and up to 70% over orthogonal fingerprinting. When the colluders scatter over tens of groups and thus the collusion pattern does not match the group structure, non-adaptive detection scheme does not have advantage over orthogonal fingerprinting, as indicated in Fig. 4.11(c). In contrast, our proposed adaptive detection is able to adjust the threshold accordingly, and thus outperforms the non-adaptive detection by $5\% \sim 10\%$ improvement on P_d. To further demonstrate the advantage of the proposed adaptive detection,



(a)

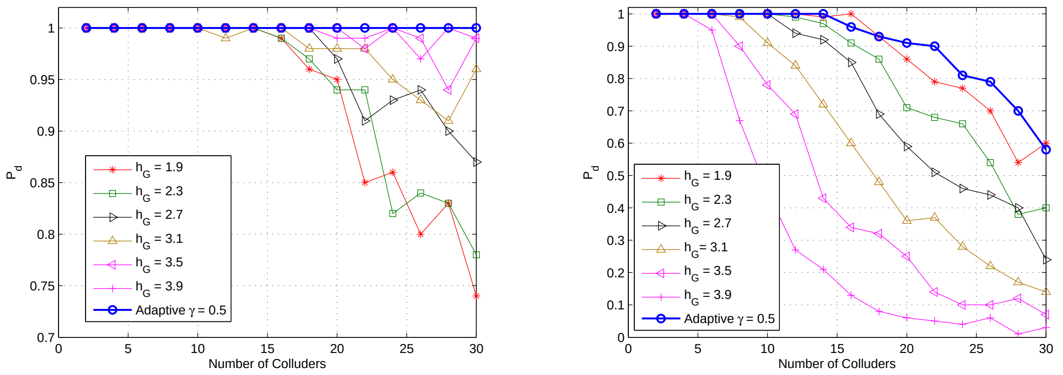
(b)

(c)

(d)

Figure 4.11: ROC curve comparisons for the adaptive and non-adaptive group-based fingerprinting. Averaging collusion attack from: (a) 2 groups, (b) 8 groups and (c) 24 groups. (d) Probability of detection vs. different collusion patterns for adaptive detection with $\gamma = 0.5$ and non-adaptive detection with α_1 of 10^{-6} to 10^{-1} under $P_{fa} = 0.01$.

we examine different collusion patterns and show the results of detection for both adaptive and non-adaptive schemes in Fig. 4.11(d). In this figure, we fix the probability of false alarm at 0.01 and examined various thresholds for non-adaptive scheme. The γ for the adaptive scheme is set at 0.5. We can see that the proposed adaptive detection consistently has better or comparable performance to non-adaptive scheme under a variety of collusion scenarios, and the performance curve of the proposed adaptive detection is the upper envelope of those achievable by non-adaptive detection at various settings.



(a) (b)

Figure 4.12: Performance comparison of adaptive detection and non-adaptive detection on Group-based ECC fingerprinting under averaging collusion attack from: (a) 2 groups, (b) distinct groups.

We have also applied the adaptive detection to coded group fingerprinting system proposed in Section 4.1 and compare it with non-adaptive detection results. Fig. 4.12 shows the detection results of the same system settings examined in Section 4.1, which is constructed based on Reed-Solomon code with alphabet size of 32 and dimension 2. The code is divided into 32 groups and each group contains 32 codewords. We vary the threshold for group detection in a wide range of 1.7∼3.9, while the adaptive detection threshold parameter γ is set at 0.5. In order to demonstrate the performance difference clearly, we choose Watermark-to-Noise-Ratio (WNR) as -10dB during the examination. Fig. 4.12(a) shows the detection under the collusion from two groups and Fig. 4.12(b) shows the results under collusion from distinct groups, i.e. any pair of colluders come from different groups. It can be seen that a fixed group detection threshold cannot generate high detection probability for various collusion patterns. A threshold of 3.9 gives satisfactory detection accuracy under 2-group collusion, but the performance drops sharply when the colluders come from distinct groups. In contrast, the proposed adaptive detection consistently leads to higher performance than the fixed thresholding scheme under various colluder numbers and collusion patterns. Overall, the adaptive detection provides performance improvement over non-adaptive detection scheme under various scenarios.

4.4 Chapter Summary

Because of cultural and other social reasons, users often form a collusion group in a foreseeable pattern. Taking advantage of this observation and based on our previous study, we have proposed a group-based joint coding and embedding fingerprinting system. In this system, the fingerprint for each user is compact and consists of user sub-codeword and group sub-codeword, which are embedded overlappingly in host signal via spread spectrum 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 inside the extracted guilty groups. Simulation results show that the proposed fingerprinting system can provide substantial improvement over existing ECC based fingerprinting. The group information helps to limit the suspicious users to a smaller set, and leads to a higher probability of detection.

In addition, we have proposed an adaptive detection approach for group-based fingerprinting. Through analyzing the existing group-based fingerprinting, we have found that the overall detection accuracy is sensitive to the group detection threshold, and the threshold achieving good performance is closely related to the collusion pattern. Based on this observation, we propose a new adaptive detection method that automatically adjusts the group detection threshold according to the detection statistics for group information to adapt to different collusion patterns. Results show that the proposed adaptive detection is superior to the non-adaptive detection under a variety of collusion scenarios, and can provide up to 10% improvement on the overall probability of detection.

Chapter 5

Collusion-Resistant Video Fingerprinting for Large User Group

With the advances of broadband communication and compression technologies, an increasing amount of video will be shared among large groups of users through the Internet and other broadband channels. For example, in applications such as cable TV, the user number can be as high as 10∼100 million. The potential adversary group may involve hundreds of colluders. However, most of the existing fingerprinting schemes consider an experimental settings with user number on the order of a few thousand and a small collusion group around 10 colluders and cannot cannot reach such large user size and high collusion resistance requirements. For example, to hold 10 million users, the Boneh-Shaw scheme [9] gives a code on the order of 10^7 bits that needs 22-hour video for embedding, and it can only resist 10 users' collusion. On the other hand, the orthogonal fingerprinting can be scaled up to hold 10 million users with collusion resistance of 100, but the detection computational complexity increases linearly with the number of users and it becomes prohibitively high for large scale system. In literature of traitor tracing through cryptographic keys [39–41], the large scale of user group has been constantly considered during the scheme design. The basic idea of these works is to use codes to establish key set for each user to access the content, which shares similarity with the coded fingerprinting such as [22, 52]. When directly extended to fingerprinting application by modulating each code symbol with spreading sequence and adding the sequence to the host signal, these schemes would have rather limited collusion resistance as the case in [22, 52] because the embedding layer is not well utilized.

Table 5.1 summarizes the performance of existing fingerprinting schemes and those extended from traitor tracing schemes for a user group of 10 million with probability of miss detection on the order of 10^{-3}. From this table, we can see that to hold such a large user group, Boneh-Shaw scheme requires an extremely long host signal and has very low collusion resistance. For all other schemes, we fix the host signal length and compare their collusion resistance and detection computational complexity. We observe a low collusion resistance for traditional ECC based fingerprinting and a high detection complexity for orthogonal fingerprinting and the anti-collusion code (ACC) fingerprinting.

Different from other fingerprinting schemes, our proposed joint coding and embedding strategies built on top of the ECC fingerprinting offer much improved collusion resistance while retaining the efficiency in construction, detection and distribution of the fingerprinted signal. Such advantages of the improved ECC fingerprinting makes it attractive for video applications, especially under the challenging settings of millions of users and hundreds of colluders. The large scale system, however, introduces several issues that have not been addressed in the

Table 5.1: Performance Comparison of Existing Fingerprinting Schemes with User Number $N_u = 10$ Million

Fingerprinting Schemes	Col. Resis.	Required Sig. Len. (N)	Det. Complexity
Boneh-Shaw FP [9,78]	10	22-hour video	$O(N + N_u)$
ECC FP [22,39,52]	<10	Several minutes video	$\mathrm{O}(\sqrt[k]{N_u}(N + N_u))$
ACC FP [64]	~ 100	Several minutes video	$O(\sqrt{N_u}N + N_u)$
Orthogonal FP [71]	~ 100	Several minutes video	$O(N_u N)$
Joint Coding-Emebedding	~ 100	Several minutes video	$O(\sqrt[k]{N_u}(N + N_u))$

existing literature. First, in large scale system where millions fingerprinted copies need to be generated and distributed, the efficient fingerprint construction with low computational complexity becomes an important issue. Meanwhile, the high collusion resistance requirement places a constraint on the code construction and embedding. How to construct and embed the code to meet efficient construction and high collusion resistance is a problem. Second, the detection of the existing improved ECC based fingerprinting employs maximal correlation based detector. Ideally, this detector can provide good performance for the large scale system. However, the large number of users will result in a high computational complexity, which becomes an issue in real-world applications. Therefore, we need to pursue a more efficient detection to reduce the computational complexity and to retain the good detection performance.

In this chapter, we explore the application of the joint coding-embedding framework to video fingerprinting for such a large scale system and address a few major design and algorithmic issues. In particular, we first address the issue of code structure to achieve higher collusion resistance and maintain the efficient construction and distribution. We then propose an efficient detection algorithm for joint coding and embedding fingerprinting which significantly speeds up the detection while maintaining a good detection performance. To our best knowledge, this is the first work of applying embedded fingerprinting on multimedia signal with such challenging user capacity and collusion resistance requirements as tens of millions of users and hundreds of colluders [35, 37].

5.1 Large-Scale Video Fingerprinting

5.1.1 Analysis of Collusion Resistance

In a collusion-resistant fingerprinting system, we usually measure the collusion resistance terms of the probability of catching one of the true colluders. Recall from Section 2.4.3, under averaging collusion and additional additive white Gaussian noise, the vector of detection statistics T_N's defined in Eqn. (2.6) follows an N_u-dimensional Gaussian distribution:

$$\mathbf{T} = [T_N(1), ..., T_N(N_u)]^T \sim N([\mathbf{m}_1, \mathbf{m}_2]^T, \sigma_d^2 \Sigma) \tag{5.1}$$
$$\text{with} \quad \mathbf{m}_1 = \|\mathbf{s}\| \left(\frac{1}{K} + \left(1 - \frac{1}{K}\right) \rho \right) \mathbf{1}_K, \quad \mathbf{m}_2 = \|\mathbf{s}\| \rho \mathbf{1}_{n-K}.$$

Here $\mathbf{1}_k$ is an all one vector with dimension k-by-1; ρ is the pair-wise correlation between fingerprint sequences; Σ is an N_u-by-N_u covariance matrix whose diagonal elements are 1's and off-diagonal elements are ρ's; σ_d^2 is the variance of the additive 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 K and fingerprint strength $\|\mathbf{s}\|$, the mean correlation values with colluders and with innocents are separated more widely for a smaller correlation ρ between each pair of sequences. This suggests that in absence of any prior knowledge on collusion patterns, a smaller ρ leads to a higher colluder detection probability P_d.

To facilitate the study, we derive the expression for the detection probability P_d based on the model in Eqn. (5.1) and the results on order statistics for multivariate Gaussian variables [63]:

$$\begin{aligned} P_d = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} & (1-\Phi^K(\frac{(t-m_1)/\sigma+\sqrt{\rho}z}{\sqrt{1-\rho}}))\times \\ & \frac{n-K}{\sigma\sqrt{1-\rho}}\Phi^{n-K-1}(\frac{(t-m_2)/\sigma+\sqrt{\rho}y}{\sqrt{1-\rho}})\times \\ & \phi(\frac{(t-m_2)/\sigma+\sqrt{\rho}y}{\sqrt{1-\rho}})\phi(y)\phi(z)dydzdt, \end{aligned} \tag{5.2}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the *c.d.f.* and *p.d.f.* of standard Gaussian distribution $N(0,1)$, respectively. The detailed derivation is presented in Section 5.5.1. We numerically examine the P_d under different ρ values for a system with $N_u = 1024$ and $N = 30,000$ under WNR$= -10$dB, and show the results in Fig. 5.1. The results are consistent with the conjecture that a small ρ value leads to a higher detection accuracy and thus is preferred in the system design. According to Section 2.4.3, the correlation ρ between fingerprint sequences constructed from ECC code with code length L, dimension k and minimum distance D can be approximated as

$$\rho \approx \frac{L-D}{L} = \frac{k-1}{L}. \tag{5.3}$$

We can choose k and L to make ρ close to 0 for good collusion resistance.

With this theoretical model, we can numerically examine the system's performance to determine whether the coded fingerprinting can meet the challenging requirements on user number and collusion resistance. In the examination, we set the total number of users $N_u = 10^8$, embeddable host signal length $N = 2\times 10^7$,

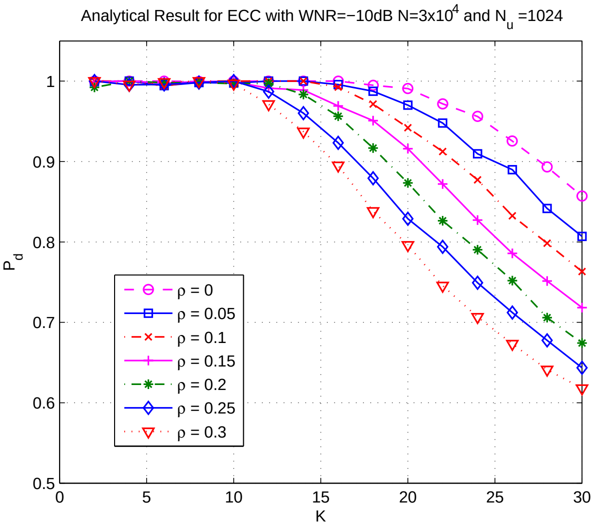


Figure 5.1: P_d of ECC fingerprinting with different intra-fingerprint correlation ρ.

which corresponds to a video signal less than 10 minutes, and correlation $\rho = 0.03$. The performance of the coded fingerprinting under this setting is shown in Fig. 5.2, where we consider different values of colluder number K with WNR (Watermark-to-Noise Ratio) ranging from -20dB to 0dB. The settings of WNR include the scenarios from severe distortion to mild distortion. We can see that even under severe distortion, WNR $= -20dB$, the system has the potential to protect a video signal as short as 10 minutes and resist more than 100 users' collusion out of 100 million users. This promising result inspires us to explore the application of ECC fingerprinting onto video signals with large user group and address several important issues in this chapter.

5.1.2 Fingerprint Construction for Reaching Large Scale

The high data volume in a video stream provides seemingly abundant spaces for data embedding and offers high degrees of freedom in choosing how to fingerprint. We need to determine how to apply ECC fingerprinting onto video signals to

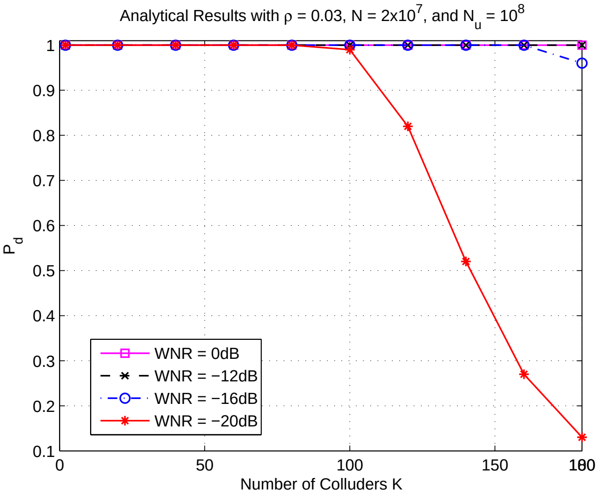


Figure 5.2: Analytical results of ECC based fingerprinting for $N_u = 10^8$, $N = 2 \times 10^7$ and $\rho = 0.03$.

achieve the large user scale. One way is to construct the first round of ECC-based fingerprint sequences as basis sequences and employ another layer of fingerprint code to reach the user capacity as shown in Fig. 5.3(a). The fingerprint sequences of the inner layer serves as the alphabet for the outer code. For example, we build an ECC based fingerprinting with Reed-Solomon code $(L, k, D)_q = (6, 2, 5)_8$ and obtain 64 fingerprint sequences according to the ECC fingerprint construction and applying permuted subsegment embedding on these sequences; then using these 64 sequences as basis sequences, we apply an outer code of RS $(62, 2, 61)_{64}$ on top of the ECC fingerprinting, arriving at an overall system for a total of 64^2 users. This scheme provides a more efficient fingerprint construction than the inner ECC fingerprint alone since it requires fewer spreading sequences due to one more code layer and also enables an efficient distribution to users. However, through the performance evaluation of ECC fingerprinting in Chapter 2, we can see that in multi-level fingerprinting schemes, the outer level code structure limits the collusion resistance of the whole system, even though its inner level has high collusion resistance. In this case, the attackers may apply segment-wise interleaving attack, i.e. one segment of the colluded signal (corresponding to one symbol in the outer code) comes from one of the colluders, which is equivalent to attackers' applying interleaving attack on the code level. Thus, this kind of fingerprinting systems mainly relies on the code level collusion resistance, which is usually very low (less than 10) due to finite alphabet size of the code. To verify this, we build a two-layer system with inner $\mathrm{RS}(6,2,5)_8$ along with permuted subsegment embedding and an outer code $\mathrm{RS}(62,2,61)_{64}$. The simulation results under interleaving collusion in Fig. 5.3(c) show that a dozen colluders are able to defeat the system even under high WNR.

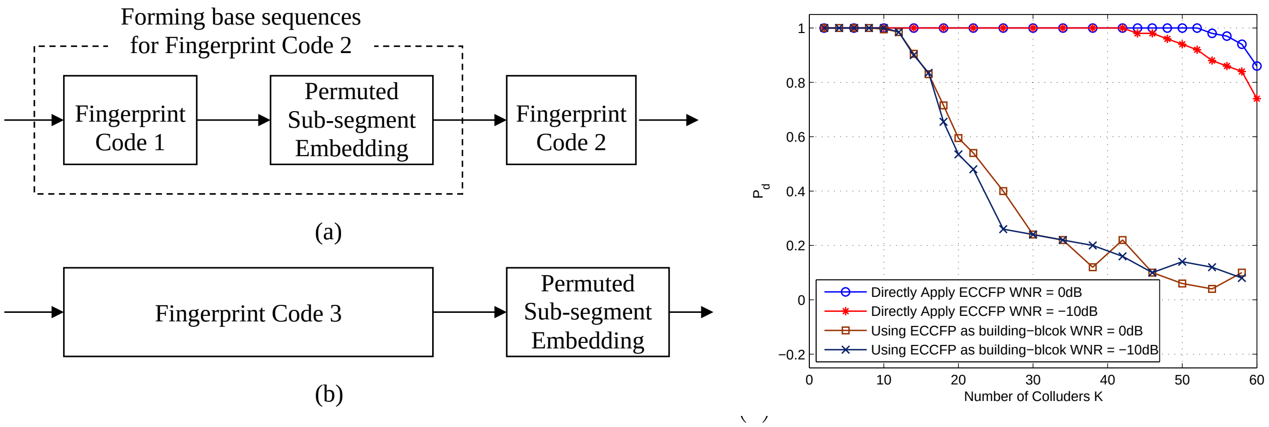


Figure 5.3: Coding-embedding structures and performance comparison. (a) Building an outer code on ECC fingerprinting to reach user capacity; (b) Using fingerprint code to reach user capacity and then applying permuted subsegment embedding; (c) Collusion resistance under interleaving collusion for both schemes in (a) and (b).

Another way is to directly apply ECC fingerprinting, that is, to first build a fingerprint code to reach the user capacity and then apply the permuted subsegment embedding to embed the fingerprint into the video signal, as shown in Fig. 5.3(b).

Continuing with the settings in the above example, this second approach would first construct a concatenated code using $\text{RS}(6, 2, 5)_8$ and $\text{RS}(62, 2, 61)_{64}$, followed by applying permuted subsegment embedding. The results are also shown in Fig. 5.3(c), where we can see a much better collusion resistance compared with the previous method. The resistance against interleaving collusion is increased from 10 colluders to more than 50 colluders under WNR of 0dB and to 40 colluders under WNR $= -10$dB. The improvement is due to the step of random permutation before the embedding. The random permutation prevents the attackers from knowing which subsegment corresponds to which symbol. As a result, the attackers cannot identify the code level and arbitrarily manipulate each symbol to mount the symbol-wise interleaving attack on either inner code or outer code. Leveraging the embedding layer, the fingerprinting system is able to resist much more colluders than the code level alone. Therefore, designing fingerprint code to reach user capacity first and then applying permuted subsegment embedding to instill randomness to enhance collusion resistance is a preferred way to construct fingerprint signals for a large number of users.

5.1.3 Efficient Detection through Trimming

Recall from Section 2.4.1 that the computational complexity of the detection for ECC-based fingerprinting is the sum of two terms: qN for demodulation and N_uL for colluder identification. When the number of users scales up to millions, the second term N_uL becomes a non-trivial part of the total computational complexity. In this subsection, we exploit code structure to significantly reduce the computational complexity.

Trimming Based Detection

We propose to employ a trimming process based on the detection results on predefined symbol positions, which we refer to as *trimming positions*. We first calculate the correlation statistics T^s_{ij} for the segments Ψ corresponding to trimming positions with every possible spreading sequence. That is:

$$T^s_{ij} = \frac{(\mathbf{z}_i - \mathbf{x}_i)^T \mathbf{u}_j}{\|\mathbf{u}_j\|}, \quad i \in \Psi; \quad j = 1, 2, ..., q.$$

where $\mathbf{z}_i$ and $\mathbf{x}_i$ are the i^{th} segment of the test signal and original signal, respectively, and $\mathbf{u}_j$ is the spreading sequence for symbol j. Then, for each trimming position $i \in \Psi$, we pick the symbols that have higher statistic than a threshold h as candidate symbols:

$$S_i = \{j | T^s_{ij} > h\}, \quad i \in \Psi. \tag{5.4}$$

The codewords that match candidate symbols in S_i for all the positions in Ψ are put into a suspicious codeword set W:

$$W = \{w | w_i \in S_i, i \in \Psi\}.$$

Finally, we apply matched filter detection of Eqn. (2.6) within the suspicious set W to identify the colluder.

The computational complexity of this scheme is determined by the number of trimming positions. If k' symbol positions are used for trimming, the resulting computational complexity for colluder identification can be reduced from $O(q^k L)$ to $O(q^{k-k'} L)$, and the reduction is $q^{k'}$-fold. For example, in a system holding around 1000 users with $q = 32$ and $k = 2$, when we use all the information symbols for trimming, we can obtain more than 1000 times reduction on the computational complexity. The detection computational complexity can be more significantly reduced for a large scale system with large q and k'.

Choosing Trimming Positions

In order to find out the corresponding codeword given the symbol values at certain positions, i.e. to find out W given $\{S_i, i \in \Psi\}$, we generally need to solve the equation of $\mathbf{x}H^T = 0$ that contains $L-k$ equations and $L-k'$ unknowns. Here, H is the parity check matrix of the code, $\mathbf{x}$ is the codeword vector with known symbols at position Ψ and unknowns at remaining positions, and $k' \leq k$. The number of solutions for these equations is $q^{k-k'}$ corresponding to the $q^{k-k'}$ suspicious codewords after the trimming. The computational complexity of solving such an equation array is $O((L-k)^3 + q^{k-k'}L)$, whose two terms correspond to Gaussian elimination process and the enumeration of all $q^{k-k'}$ codewords satisfying the equations, respectively. To further reduce the complexity, we employ systematic construction for Reed-Solomon code to build fingerprint code and use the information symbols for trimming. In the systematic code construction, the first k symbols in the codeword are the information symbols, the remaining are parity check symbols. These information symbols provide the index of users and can be used to easily identify the users/codewords. The position information of these symbols (or the corresponding segments) is protected from adversaries by the random permutation during the permuted subsegment embedding. To achieve higher detection accuracy, it is desirable to assign more energy on the trimming symbols. We accomplish this by expanding the fingerprint sequence length by γ times for each information symbol, so that the segment length becomes $N_s = \gamma N/(L + k(\gamma - 1))$, where N is the total sequence length and L is the codeword length. The segment size for remaining symbols would be $N/(L+k(\gamma-1))$. The expansion can be implemented by repeatedly embedding the sequence corresponding to the information symbols by γ times.

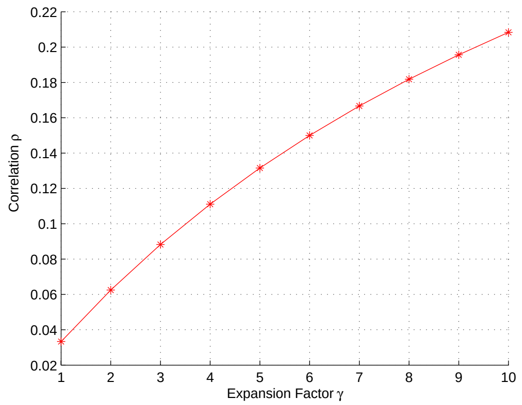


Figure 5.4: Correlation ρ vs. Expansion factor γ.

One of the effects of the expansion is the increase in correlation between fingerprint sequences. Recall that a Reed-Solomon code with parameters (L, k) has the minimum distance of $D = L - k + 1$ or any two codewords share at most $k - 1$ symbols. For some pairs of codewords, the symbols in common lie at the information symbol positions. Thus after the expansion, the number of shared symbol becomes $\gamma(k-1)$. For codewords that have fewer than $k - 1$ information symbols in common, the number of shared symbols after the expansion would be smaller than $\gamma(k-1)$. Therefore, in the expanded code, the minimum distance becomes $L + k(\gamma - 1) - \gamma(k-1) = L - k + \gamma$, and the maximal correlation of the fingerprint sequences would become

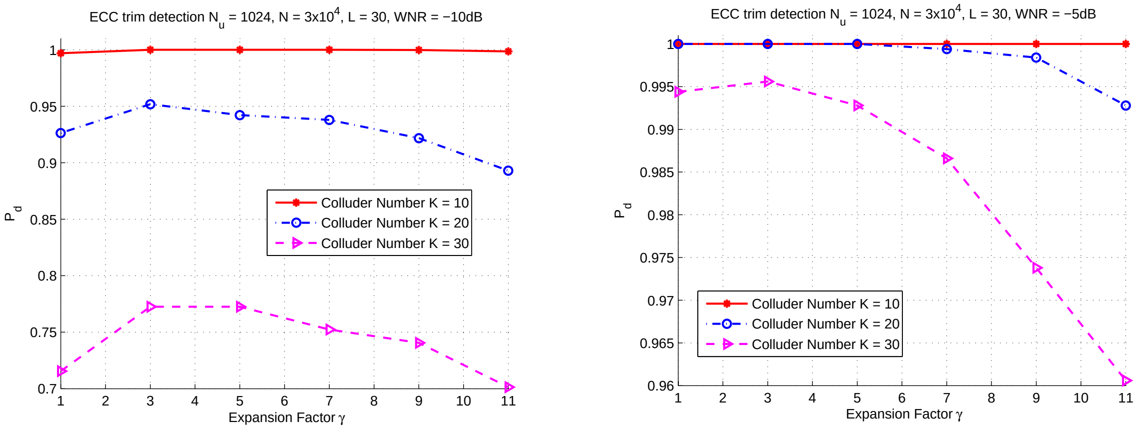
$$\rho' = \frac{\gamma(k-1)}{L + k(\gamma - 1)}. \tag{5.5}$$

Fig. 5.4 shows the relationship between the correlation ρ' and the expansion factor γ. We can see that the correlation increases as γ becomes larger, and in turn the overall detection accuracy may decrease according to the results in Eqn. (5.1). On the other hand, high γ enhances the accuracy of trimming symbol detection and thus may lead to a high overall probability of detection P_d. We examine the effect of the expansion factor γ on P_d and show the results in Fig. 5.5 at WNR of $0dB$

and $-10dB$. In this figure, we set the host signal length as $N = 30,000$ and the total number of users $N_u = 1024$. The code is constructed by Reed-Solomon code with $L = 30, k = 2$ and $q = 32$. Both information symbols are used for trimming. From the results, we find that there is an optimal value of γ for a given k' to achieve the highest detection probability. For the particular experimental settings in Fig. 5.5, $\gamma = 3$ achieves the best detection results. In the next section, we will theoretically analyze P_d as a function of the system parameters, and optimal γ can be derived based on the theoretical model. In Fig. 5.6, we examine P_d of trimming detection and matched-filter detection with $\gamma = 3$ under WNR $= -10dB$ and $-5dB$. We can see that, compared with matched-filter detection, the accuracy of trimming detection is only reduced by less than 6% for low WNRs and less than 0.5% for moderate WNRs. In most fingerprinting applications, since the host signal is usually available to the detector and its interference can be removed from the test signal, we expect a WNR higher than -5dB and thus a comparable performance by trimming detection to matched-filter detection. Meanwhile, the trimming detection reduces the detection computational complexity by $q^{k'}$, which is around 1000 times in this experimental settings. As can be seen from the results, the efficiency of trimming detection is at a negligible expense of lower performance than matched-filter detection

Performance Analysis

We analyze the performance of the proposed trimming detection in terms of the detection probability. For simplicity, we show the analysis for the case that only one information symbol position is used for trimming and the symbol with highest correlation statistic is chosen to trim the codebook. Thus after trimming, there

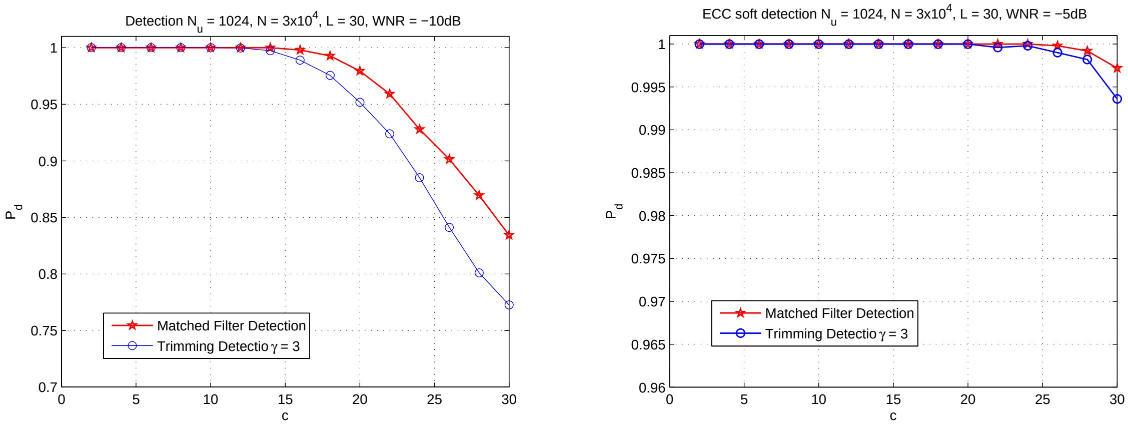


(a) (b)

Figure 5.5: P_d vs. expansion factor γ by trimming detector at WNR of (a) -10dB and (b) -5dB.

remain q^{k-1} codewords in the codebook. We assume random collusion in our analysis, i.e. all the users have equal probability to participate the collusion.

Without loss of generality, we select the first symbol position as the trimming position and its corresponding frame size is $N_s = \gamma N/(L + k(\gamma - 1))$. We call the symbol in the alphabet as *color* and thus there are q colors for a q-ary codebook. Note that for a linear code, such as Reed-Solomon code considered in this chapter, at any given position, each color occurs the same number of times among all the codewords. That is, each color occurs in q^{k-1} codewords among totally q^k codewords. When considering random collusion attack, the effect of forming a K-colluder group on the first symbol position is to choose K symbols from q^k symbols $\{11...122...233...3......qq...q\}$. The total number of possibilities for such a K-colluder group are $\binom{q^k}{K}$. Denote the colluders' color pattern at the trimming position as a vector $\mathbf{c} = [c_1\ c_2\ ...\ c_n]$ and $\sum_{i=1}^{n} c_i = K$, where n is the total number of colors and c_i is the number of symbols with color i. For example, a vector $\mathbf{c} = [1\ 2\ 2]$ means that among the first symbol position of 5 colluders' codewords,



(a)

(b)

Figure 5.6: Performance comparison of trimming detection vs. matched filter detection with expansion factor $\gamma = 3$ at WNR of (a) -10dB and (b) -5dB.

there are three different colors in total: one color appears once, one appears twice, and another also appears twice. The exact color values do not affect the analysis here, because of the symmetry of the code as well as the randomness of the collusion and the mutual independence of the fingerprint sequences for different colors. For example, for the color pattern [1 2 2] and color alphabet $\{A, ..., F\}$, the instance of $\{A\ A\ B\ D\ D\}$ and $\{A\ B\ B\ F\ F\}$ would result in the same detection probability. Given this vector $\mathbf{c}$, we can derive the probability of detection as:

$$\begin{aligned} P_{d|\mathbf{c}} &= \sum_{i=1}^{n} Pr(\text{catch one colluder inside subcode } i) \times Pr(\text{subcode } i \text{ is picked}) \\ &= \sum_{i=1}^{n} P_d(c_i, q^{k-1}) Pr(T_i = \max(T_1, .., T_q)). \end{aligned}$$

Here, "subcode i" refers to the set of codewords having color i at the trimming position; $P_d(c_i, q^{k-1})$ is calculated according to Eqn. (5.2) with K replaced by c_i and n replaced by q^{k-1}; and Eqn. (5.5) is used for calculating ρ'. The probability of subcode i being picked at the trimming position is calculated by

$$Pr(T_i = \max(T_1, .., T_q)) = \int_{-\infty}^{\infty} \Pi_{j \neq i} Pr(T_j < t) f_i(t) dt,$$
$$T_j \sim N(\frac{c_j}{K}\|\mathbf{u}\|, \sigma_d^2) \quad j = 1, ..., q \tag{5.6}$$

where T_j is the detection statistic for symbol j and $f_i(t)$ is the $p.d.f.$ of T_i. Note that T_j's are independent Gaussian variables with equal variance, due to the presence of white Gaussian noise and the orthogonality of the sequences representing different symbols. For those colors that are not contained in the colluders' codewords, c_j's are set to 0.

After obtaining $P_{d|\mathbf{c}}$, we can calculate the overall probability of detection as

$$P_d = \sum_{\mathbf{c} \in \mathcal{C}_K} P_{d|\mathbf{c}} \times Pr(\mathbf{c}). \tag{5.7}$$

Here, $\mathcal{C}_K$ is the set of all color patterns for K colluders, and it can be recursively expressed in a matrix form

$$\mathcal{C}_K = \begin{bmatrix} \mathbf{1} & \mathcal{C}_{K-1} \\ \mathbf{2} & \mathcal{C}_{K-2}^{\backslash\{1\}} \\ \cdots & \cdots \\ \mathbf{i} & \mathcal{C}_{K-i}^{\backslash\{1,\ldots,i-1\}} \\ \cdots & \cdots \\ K & \mathbf{0} \end{bmatrix},$$

where $\mathcal{C}_K^{\backslash\mathcal{I}}$ denote the set of row vectors in $\mathcal{C}_K$ excluding the rows containing element in $\mathcal{I}$. $\mathbf{1}$ represents an all "1" column vector with the same number of entries as the row number of $\mathcal{C}_{K-1}$, and $\mathbf{2}$ represents an all "2" column vector with the same number of entries as the row number of $\mathcal{C}_{K-2}^{\backslash\{1\}}$, and so on. In the above matrix, each row vector contains q elements and represents one possible color pattern generated from K colluders. For each color pattern $\mathbf{c}$ in $\mathcal{C}_K$, it can be shown that

$$\begin{aligned} Pr(\mathbf{c}) &= \frac{\theta \prod_{i=1}^{q} \binom{q^{k-1}}{c_i}}{\binom{q^k}{K}} & (5.8) \\ \text{with} \quad \theta &= \binom{q}{n} \binom{n}{a_1, a_2, ..., a_m} = \binom{q}{n} \frac{n!}{\prod_{i=1}^{m} a_i!}. & (5.9) \end{aligned}$$

Here, $\mathbf{a} = [a_1, a_2, ..., a_m]$ is an auxiliary vector representing the histogram of a color pattern $\mathbf{c}$, m is the number of distinct values in vector $\mathbf{c}$ and a_i is the number of occurrence of ith value in vector $\mathbf{c}$. The detailed derivations for $\mathcal{C}_K$ and $Pr(\mathbf{c})$ are presented in Section 5.5.2.

According to the above analysis, we are able to derive the $\mathcal{C}_K$ matrix for any K value and get the α value for each $\mathbf{c}$ vector (each row of $\mathcal{C}_K$). By plugging these quantities into Eqn. (5.7), we obtain the analytical expression of probability of detection P_d for the trimming method. The optimal parameter γ^* can be chosen to maximize P_d so that

$$\frac{\partial P_d(\gamma)}{\partial \gamma}|_{\gamma=\gamma^*} = 0.$$

The analysis for a more general trimming detection can be conducted in a similar way. For example, we can set a threshold for the trimming detection statistic T to select multiple symbols as shown in Eqn. (5.4). The only changes that need to be made is the calculation of $P_{d|\mathbf{c}}$, which now becomes

$$\begin{aligned} P_{d|\mathbf{c}} = & \sum_{i=1}^{n} P_d(c_i, q^{k-1}) Pr(T_i > h) \prod_{m=1, m\neq i}^{q} Pr(T_m < h) + \\ & \sum_{i=1}^{n} \sum_{j=n+1}^{q} P_d(c_i, 2q^{k-1}) Pr(T_i > h) Pr(T_j > h) \prod_{m=1, m\neq i,j}^{q} Pr(T_m < h) + \\ & \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} P_d(c_i + c_j, 2q^{k-1}) Pr(T_i > h) Pr(T_j > h) \prod_{m=1, m\neq i,j}^{q} Pr(T_m < h) \\ & + ... + P_d(K, nq^{k-1}) \prod_{m=1}^{n} Pr(T_m > h) \prod_{m=n+1}^{q} Pr(T_m < h), \qquad (5.10) \end{aligned}$$

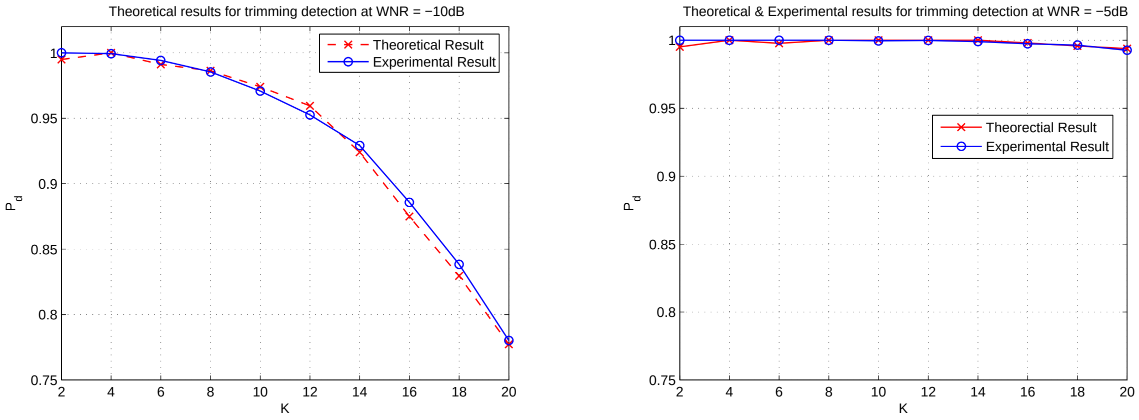
where h is a threshold, and

$$Pr(T_i > h) = Q(\frac{h - c_i\|\mathbf{u}\|/K}{\sigma_d})$$

with

$$Q(t) = \int_t^\infty 1/\sqrt{2\pi}\exp(-x^2/2)dx.$$

Multiple-position trimming can be analyzed by iteratively applying the above analysis process.



(a) (b)

Figure 5.7: Comparison of theoretical and experimental results of trimming detection: (a) -10dB and (b) -5dB.

We numerically examine the analysis result with the parameter settings of $N_u = 1024$, $q = 32$, and $k = 2$. The trimming detection is performed only on the first symbol position and the color with highest detection statistic is picked for trimming. The results for two WNR values (-10dB and -5dB) are shown in Fig. 5.7, along with the simulation results at the same parameter settings. We can see that the analytical results match the experimental results well for most of the K values. The small gap at some points comes from the numerical computation error on the combination terms in $Pr(\mathbf{c})$ and the calculation of $P_{d|\mathbf{c}}$, where a simplification of equal-correlation is assumed in the analysis while the actual fingerprints have different correlation values according to the codebook construction.

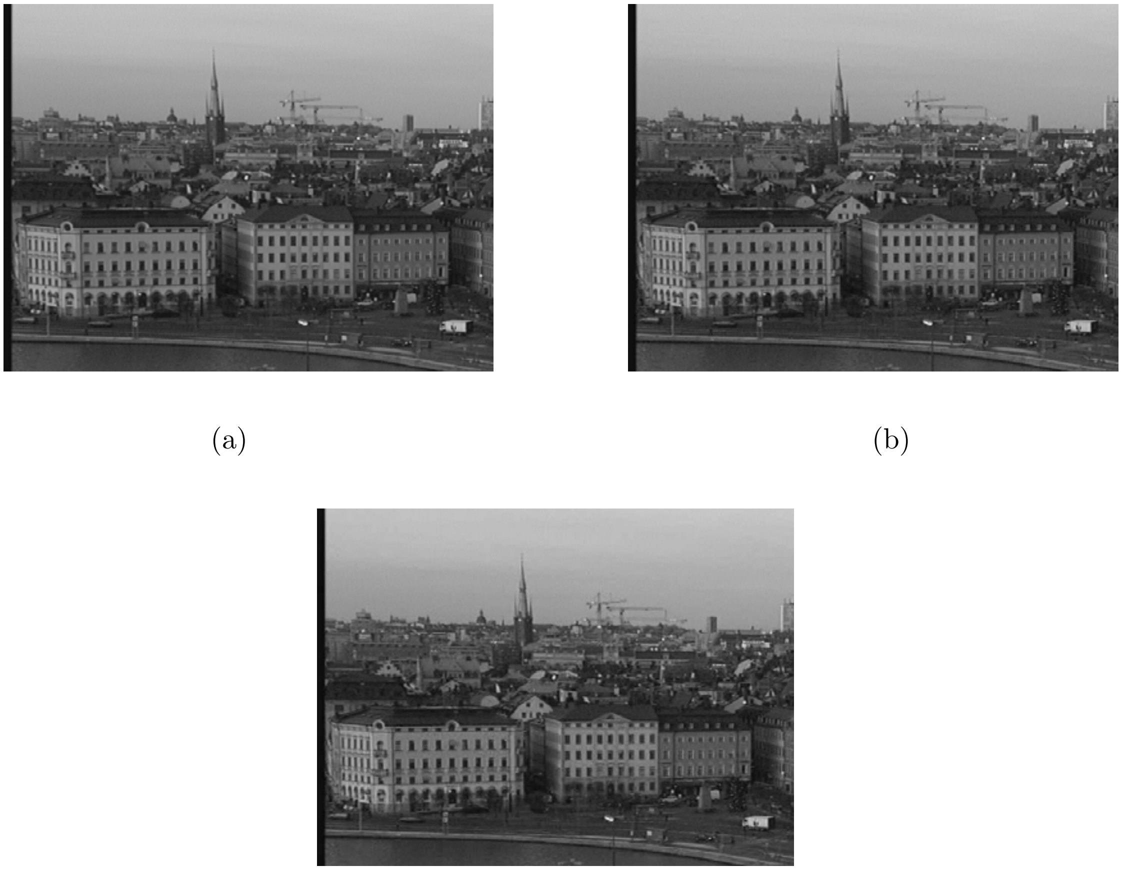
5.2 Experimental Results

5.2.1 Experimental settings and results

In this section, we apply the proposed fingerprinting scheme on video signals and examine the experimental performance. The test video signal is obtained from [3] and has VGA size of 640-by-480. The total number of users that we target at is on the order of ten million and colluder number is around 100. We choose a Reed-Solomon code with $q = 64$, $k = 4$, and $L = 63$, which leads to the number of users $N_u = 1.6 \times 10^7$. The expansion parameter γ in the efficient detection is set at 3, and the first 4 symbol positions are selected for trimming. Thus the equivalent codeword length is 71 and the resulting pairwise maximum correlation is $\rho' = 0.127$ according to Eqn. (5.5).

During the fingerprint embedding, each frame is transformed into DCT domain. Fingerprint sequences are embedded into these DCT coefficients through additive embedding with perceptual scaling. The host video signal is chosen to have 852 frames with about 12 frames for each codeword symbol. For simplicity, we repeatedly embed the same fingerprint sequence into every group of frames consisting of 6 consecutive frames [77]. The issue of intra-video collusion attack will be discussed in Section 5.3.2. Subsegment partition factor β is set as 24. Fig. 5.8(a)-(c) show the 500^{th} frame in the original, fingerprinted and compressed video sequences. Fig. 5.9 shows the simulation results on the probability of catching one colluder,

P_d, versus colluder number, K, under averaging and interleaving collusion attacks followed by MPEG compression. The curves are obtained by averaging the results of 50 iterations.



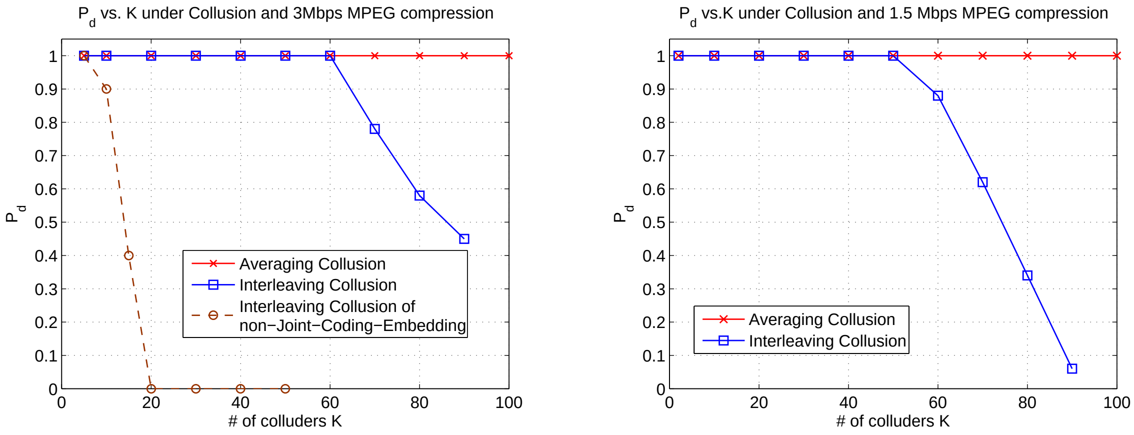
(a)

(b)

(c)

Figure 5.8: Experimental results: (a) Original frame; (b) Fingerprinted frame before attack with PSNR = 32dB; (c) 3Mbps MPEG compressed frame.

The results shown in Fig. 5.9 is encouraging in that we are able to hold more than 10 million users and resist more than 100 users' averaging collusion and 60 users' interleaving collusion within less than 30 seconds video. The resistance can be further improved by increasing the video sequence length and employing a



(a)

(b)

Figure 5.9: Probability of catching one colluder P_d vs. colluder number c under averaging and interleaving collusion followed by (a) 3M bps and (b) 1.5M bps MPEG-2 compression.

larger β value, and trading off the reduced efficiency in distributing fingerprinted signals. On the other hand, without the joint coding and embedding approach, the system can only resist about 2 users' collusion as indicated by the dash line in Fig. 5.9(a). We can see that the joint coding and embedding strategy can help overcome the code-level limitation and substantially improve the collusion resistance at an affordable computational complexity. We further decrease the compression bit rate down to 1.5Mbps which is of VCD quality and examine the collusion resistance. Even under this quality, the collusion resistance only reduces a little under interleaving collusion down to 50 colluders. The resistance under averaging collusion is till higher than 100 colluders.

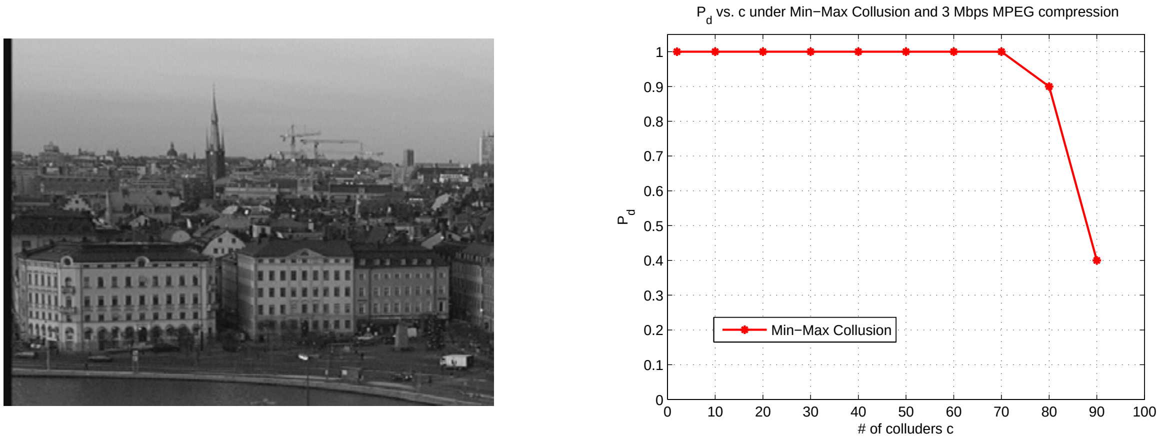
5.2.2 Results under Nonlinear Collusion Attack

The experimental and analytical results that we have presented so far are based on averaging collusion and interleaving collusion. Our next experiments examine the collusion resistance under Min-Max attack which can be used as a representative of non-linear collusion[1]. In the Min-Max attack, colluders choose the average of the minimum and maximum values of their copies in each DCT coefficient position to generate the colluded version. MPEG-2 compression is further applied to the colluded signal. Fig. 5.10(a) shows the 500^{th} frame of the colluded video after compression and Fig. 5.10(b) shows the detection probability P_d versus the colluder number under the Min-Max attack followed by 3Mbps MPEG-2 compression. The results show that under this non-linear collusion attack, the collusion resistance is around 80 colluders, which further demonstrates the effectiveness of the proposed large scale fingerprinting. Its performance gap compared with that under averaging collusion is mainly because the Min-Max collusion introduces higher distortion on the colluded signal than the averaging collusion [82]. For 80 users' collusion, the Mean Squared Error (MSE) introduced to the host signal by averaging attack is 0.94 and is 3.57 by Min-Max attack before MPEG-2 compression. The increased distortion results in a lower collusion resistance of the system to Min-Max collusion attack.

It is worth mentioning that the computational complexity of order-statistics based non-linear collusion significantly increases because of the sorting involved. In the examined experimental settings, the non-linear collusion attack from 80 colluders requires 12 hours while the averaging collusion only needs 70 minutes. This

[1] Interested readers may refer to [82] for detailed analysis on the relationship and comparison among various nonlinear collusion attacks.

high computational complexity can also help deter the colluders from employing the non-linear collusion on video especially for a large colluder group.



(a)

(b)

Figure 5.10: Experimental results under Min-Max collusion followed by 3Mbps MPEG2 compression: (a) 500th frame after attack; (b) Probability of catching one colluder P_d vs. colluder number c.

Table 5.2: Time Consumption of the Proposed Efficient Fingerprinting Embedding and Detection with User Number N_u = 16 Million and 852 VGA Frames

	Efficient Scheme	Fully Embedding/Matched Filter Detection
Embedding (per copy)	4–5 mins	30 mins
Detection	370 secs	840 secs

5.2.3 Performance Summary

We summarize the time consumption of the embedding and detection in Table 5.2. The efficient scheme for embedding refers to the way that we generate all the possible versions beforehand for each subsegment and concatenate these subsegments according to each user's fingerprint code. Fully embedding means we perform the embedding on the entire host signal for every user without exploring the code structure. The efficient scheme for detection refers to the trimming detection proposed in Section 5.1.3. Standard compilation from Microsoft Visual Studio has been applied to the C++ implementations of both schemes. We can see from Table 5.2 that the code structure of the fingerprinting speeds up the fingerprinted signal generation by 6-7 times, and the proposed trimming detection only consumes less than half of the time required by the matched-filter detection.

5.3 Discussions

5.3.1 Relation to Group-based ECC Fingerprinting

The proposed trimming detection utilize the code structure to first detect trimming symbols and then use these symbols to trim the codebook. This process is similar to the detection of GRACE fingerprinting proposed in Chapter 4. In this group-based fingerprinting, the group symbols are first extracted and then only codewords inside the extracted groups are put under suspicion for further examination. However, the GRACE fingerprinting cannot be applied here to get efficient detection because of the following reasons:

First, GRACE fingerprinting spreads group information over the entire signal, which does not allow the efficient construction of fingerprinted signal. Recall that the efficiency of the joint coding and embedding fingerprinting lies in the code structure such that many copies share the same segment. By pre-generating those segments, we can assemble the fingerprinted signal for each user according to his/her codeword to meet real-time requirement. However, in GRACE fingerprinting, the group information is spread over the entire signal and is different for users from different group. After adding the group information, users have few segments in common, thus we cannot utilize the efficient construction as before.

Second, the group fingerprinting can be implemented by appending the group information to maintain the efficiency in fingerprint construction and to employ permuted sub-segment embedding hoping to protect the group information. This scheme, however, is vulnerable to the group-framing attack discussed in Section 4.1.4 because colluders can identify the shared sub-segments and guess the group information.

In contrast, the proposed trimming detection takes advantage of the inherent code structure and randomization in embedding, which cannot be easily identified by the colluders but can be explored by the detector to perform the efficient detection.

5.3.2 Intra-video collusion

The large amount of data in video is a double-edged sword as it also benefits the attackers. Given one copy of the fingerprinted video, an attacker may apply multiple-frame collusion [57, 58], whereby several frames are used to estimate and eventually remove the fingerprint. One possible implementation of such intra-video attacks is that the attacker may average several frames that have "visually similar" content but are embedded with independent fingerprint sequences. By collecting enough frames, this averaging operation can successfully remove the embedded fingerprint at a possible expense of reduced visual quality. Furthermore, this attack can be extended to object-based collusion, where similar objects are identified and averaged or swapped to circumvent the detection. Another possible attack is that the attacker may identify several “visually dissimilar” frames or regions embedded with the same fingerprint sequences and average these frames to estimate the embedded fingerprint. This estimated fingerprint sequence will then be subtracted from the fingerprinted signal to obtain an approximation of the original frame. In each of the above cases, the attacker can succeed by attacking just one fingerprinted copy without help from other colluders. Therefore, the design and embedding of the fingerprint sequence should be robust to these intra-video attacks.

A basic principle to resist these attacks is to embed fingerprint sequences based on the content of the video, i.e. similar fingerprint sequences are embedded in frames/objects with similar content and different fingerprint sequences are embedded in frames with different content [60]. For example, a scheme proposed by Fridrich employs the content-based hash of each segment of the signal as a key to generate watermark sequences [24]. However, this method cannot be directly used in the ECC-based fingerprinting because of two reasons. First, the fingerprint spreading sequences in ECC fingerprinting are mainly determined by the code structure and the designer does not have the freedom to generate the sequences according to the content. In ECC-based fingerprinting, the spreading sequences for different symbols should be orthogonal to each other. On the other hand, the content-based construction of fingerprint sequences would lead to correlated spreading sequences for different symbols, which conflicts with the fingerprint construction for ECC fingerprinting. Second, the correlation between the watermark sequences for two frames generated in [24] decrease exponentially as the Hamming distance between their hash values increases. This change is so dramatic that it may result in visually similar frames having quite different watermarks. Therefore, to address the intra-video collusion attack, we need to find a mechanism that is

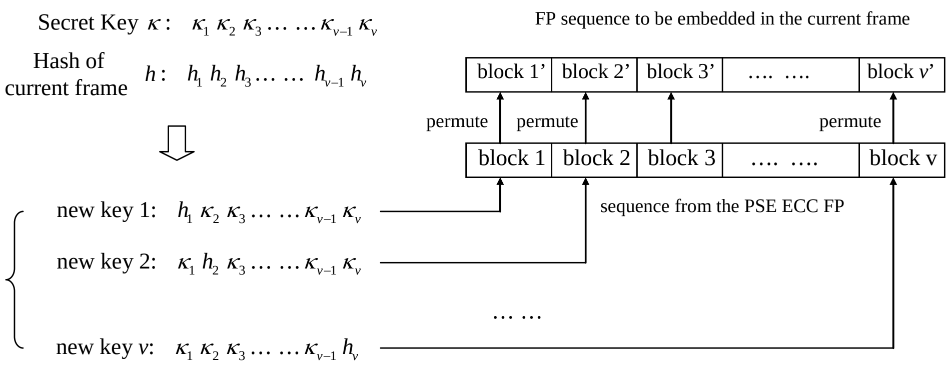

Secret Key κ: $\kappa_1\ \kappa_2\ \kappa_3 \ldots \ldots \kappa_{v-1}\ \kappa_v$
Hash of current frame
h: $h_1\ h_2\ h_3 \ldots \ldots\ h_{v-1}\ h_v$
new key 1: $h_1\ \kappa_2\ \kappa_3 \ldots \ldots \kappa_{v-1}\ \kappa_v$
new key 2: $\kappa_1\ h_2\ \kappa_3 \ldots \ldots \kappa_{v-1}\ \kappa_v$
new key v: $\kappa_1\ \kappa_2\ \kappa_3 \ldots \ldots \kappa_{v-1}\ h_v$
FP sequence to be embedded in the current frame
block 1'
block 2'
block 3'
…. ….
block v'
permute
permute
permute
block 1
block 2
block 3
…. ….
block v
sequence from the PSE ECC FP
… …

Figure 5.11: Illustration of hash-based fingerprint construction.

able to convert the sequence structure imposed by the codeword to what the video content demands, and build content based fingerprints such that the difference between fingerprints of two frames is linear with respect to the difference of frame content.

Since consecutive frames in one scene are visually similar, we can repeatedly embed the same fingerprint sequence into those consecutive frames. For those visually dissimilar frames that are assigned similar fingerprint sequences based on ECC construction, we need to modify the fingerprint sequences to be dissimilar. To achieve this goal, we can use the hash of each frame to adjust the fingerprint sequences. For example, frame i and frame j have hash $h^{(i)}$ and $h^{(j)}$, respectively, each of which is a v-bit binary sequence. If frame i and frame j are visually similar, the Hamming distance between $h^{(i)}$ and $h^{(j)}$, $D(h^{(i)}, h^{(j)})$, would be very small, i.e. only a few bit positions are different; if they are visually different, $D(h^{(i)}, h^{(j)})$ will be very large. As illustrated in Fig. 5.11, we choose a binary secret key κ with the same length v as the frame hash to generate fingerprint sequence. For each frame i, we generate v new keys by replacing κ's n^{th} bit with h_i's n^{th} bit, $n = 1, ..., v$. Meanwhile, the fingerprint sequence for the current frame is divided into v blocks.

Each of these v new keys is used to permute one block. The concatenation of these v permuted blocks will be the final fingerprint sequence for frame i. With this method, visually similar frames will have many permuted blocks in common, thus the overall sequence will be similar; and visually dissimilar frames will have dissimilar sequences. The correlation between the sequences changes linearly with the Hamming distance between two frames' hashes.

5.4 Chapter Summary

In this chapter, we consider fingerprinting video signal under such challenging settings as to accommodate millions of users and to resist hundreds of users' collusion. Our work of joint coding and embedding ECC fingerprinting has shown an excellent trade-off between collusion resistance and efficient construction and detection, which is promising for large-scale video fingerprinting. Building upon this improved ECC fingerprinting, we address issues of designing code structure and speeding up detection. We have found out that directly applying the joint coding and embedding fingerprinting scheme is preferred for a good trade-off between efficient generation and collusion resistance. The proposed trimming approach can speed up the detection by 3 orders of magnitude with only slightly drop of detection accuracy. With the proposed fingerprint construction and efficient detection, the system holding 16 million users can resist 50-60 colluders' interleaving collusion and more than 100 users' averaging collusion as well as 80 users' non-linear collusion. The user capacity and collusion resistance can be further increased by adjusting such system parameters as k and β. Both the analysis and the experimental results show a strong potential of joint coding and embedding ECC fingerprinting for large-scale video fingerprinting applications.

5.5 Appendix: Derivations

5.5.1 Derivation of Detection Probability P_d of (5.2)

Based on the distribution of the detection statistic in Eqn. (5.1), we derive the probability of detection as follows. We denote the maximum detection statistic for colluder and innocent user as T_1 and T_2, respectively. That is,

$$T_1 = \max_{i \in S_C} T_i, \quad T_2 = \max_{i \notin S_C} T_i.$$

The probability of catching one colluder using the maximum detector (2.6) can be expressed as

$$P_d = Pr(T_1 > T_2) \approx \int_{-\infty}^{\infty} Pr(T_1 > t) f_{T_2}(t) dt, \tag{5.11}$$

The approximation in the above equation comes from the simplification on the independence between T_1 and T_2 due to small correlation ρ. To obtain an expression of P_d, we employ the results on order statistics from [63]: Let $\mathbf{x} = [X_1, ..., X_n]'$ have an n-dimension Gaussian distribution $N_n(\mu, \mathbf{\Sigma})$ such that the mean value μ_i, variance σ_i^2 of X_i, and the correlation coefficient ρ_{ij} of X_i and X_j satisfy

$$\mu_1 = \mu_2 = ... = \mu_n = \mu;$$
$$\sigma_1^2 = \sigma_2^2 = ... = \sigma_n^2 = \sigma^2;$$
$$\rho_{ij} = \rho \in [0, 1].$$

Then the density function and distribution function of n^{th} smallest variable $X_{(n)}$, denoted as $g_{(n)}(x)$ and $G_{(n)}(x)$, are

$$g_{(n)}(x) = \int_{-\infty}^{\infty} (\sigma\sqrt{1-\rho})^{-1} f_{(n)} \left(\frac{(x-\mu)/\sigma + \sqrt{\rho} z}{\sqrt{1-\rho}} \right) \phi(z) dz,$$
$$G_{(n)}(x) = \int_{-\infty}^{\infty} F_{(n)} \left(\frac{(x-\mu)/\sigma + \sqrt{\rho} z}{\sqrt{1-\rho}} \right) \phi(z) dz, \text{for } x \in (-\infty, \infty),$$
$$\text{where } f_{(n)}(z) = n\Phi^{n-1}(z)\phi(z), \quad F_{(n)}(z) = \Phi^n(z). \tag{5.12}$$

Here, $\Phi(\cdot)$ and $\phi(\cdot)$ denote the cumulative distribution function (*c.d.f.*) and the probability density function (*p.d.f.*) of standard Gaussian distribution $N(0,1)$, respectively. Based on this result, we can get the *c.d.f.* for T_1 and *p.d.f.* for T_2 by plugging in the proper μ value with m_1 for colluder and m_2 for innocent user. The expression for the probability of detection becomes what is shown in (5.2):

$$\begin{aligned} P_d = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} & (1-\Phi^c(\frac{(t-m_1)/\sigma+\sqrt{\rho}z}{\sqrt{1-\rho}}))\times \\ & \frac{n-c}{\sigma\sqrt{1-\rho}}\Phi^{n-c-1}(\frac{(t-m_2)/\sigma+\sqrt{\rho}y}{\sqrt{1-\rho}})\times \\ & \phi(\frac{(t-m_2)/\sigma+\sqrt{\rho}y}{\sqrt{1-\rho}})\phi(y)\phi(z)dydzdt. \end{aligned}$$

5.5.2 Derivation of $Pr(\mathbf{c})$ and $\mathcal{C}_K$

From Eqn. (5.7), we can see that to get the overall detection probability P_d, we need to obtain two quantities: the set of possible $\mathbf{c}$ patterns for K colluders, $\mathcal{C}_K$, and the probability for each pattern to occur, $Pr(\mathbf{c})$. We assume that the collusion happens randomly among a total of q^k users, and the total number of possible choices for K colluders is $\binom{q^k}{K}$. For a given color pattern $\mathbf{c}=[c_1,c_2,...,c_q]$, the number of possibilities is $\theta\prod_{i=1}^{q}\binom{q^{k-1}}{c_i}$, where θ is the number of instances of color pattern $\mathbf{c}$, and the term $\prod_{i=1}^{q}\binom{q^{k-1}}{c_i}$ is the number of choices of colluders' codeword that has c_i occurrences for each color. Then the $Pr(\mathbf{c})$ can be obtained by

$$Pr(\mathbf{c})=\frac{\theta\prod_{i=1}^{q}\binom{q^{k-1}}{c_i}}{\binom{q^k}{K}}.$$

To calculate θ, we first take a look at an example with $q=4$, and $K=7$. We use $\{A,B,C,D\}$ to denote the four colors in the alphabet. The possible instances of a color pattern $\mathbf{c}=[1\ 1\ 2\ 0]$ are $\{ABCC\}$ $\{ABBC\}$ $\{AABC\}$ $\{ABDD\}$ $\{ABBD\}$ $\{AABD\}$ $\{ACDD\}$ $\{ACCD\}$ $\{AACD\}$ $\{BCDD\}$ $\{BCCD\}$ $\{BBCD\}$, giving a total of $\binom{4}{3}\binom{3}{2}\binom{3-2}{1} = 12$ instances. For any value of q, K and $\mathbf{c}$, the value of θ can be obtained as follows. For a given color pattern $\mathbf{c} = [c_1, c_2, ..., c_n, 0, 0...0]$ leading by n non-zero elements, we derive a histogram vector $\mathbf{a} = [a_1, a_2, ..., a_m]$, where m is the number of distinct values in vector $\mathbf{c}$ and a_i is the number of occurrence of ith value in vector $\mathbf{c}$. For the above example $\mathbf{c} = [1\ 1\ 2\ 0]$, we have $\mathbf{a} = [2\ 1]$ indicating that there are two '1's and one '2' in $\mathbf{c}$. Apparently, $\sum_{i=1}^{m} a_i = n$. Then θ can be calculated by

$$\begin{aligned}\theta &= \binom{q}{n}\binom{n}{a_1}\binom{n-a_1}{a_2}\binom{n-a_1-a_2}{a_3}... \\ &= \binom{q}{n}\binom{n}{a_1, a_2, ..., a_m} = \binom{q}{n}\frac{n!}{\prod_{i=1}^{m} a_i!}.\end{aligned} \tag{5.13}$$

The next step is to derive the color patterns for a given K, denoted as $\mathcal{C}_K$. In the following, we use a matrix to represent $\mathcal{C}_K$ in which each row is a color pattern. $\mathcal{C}_K$ can be derived as follows:

For K = 1:

$$\mathcal{C}_1 = [1];$$

For K = 2:

$$\mathcal{C}_2 = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix};$$

For K = 3:

$$\mathcal{C}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 3 & 0 & 0 \end{bmatrix};$$

For K = 4:

$$\mathcal{C}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 1 & 3 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 4 & 0 & 0 & 0 \end{bmatrix};$$

For any K,

$$\mathcal{C}_K = \begin{bmatrix} \mathbf{1} & \mathcal{C}_{K-1} \\ \mathbf{2} & \mathcal{C}_{K-2}^{\backslash\{1\}} \\ \cdots & \cdots \\ \mathbf{i} & \mathcal{C}_{K-i}^{\backslash\{1,\ldots,i-1\}} \\ \cdots & \cdots \\ K & \mathbf{0} \end{bmatrix};$$

where $\mathcal{C}_K^{\backslash\mathcal{I}}$ denote the set of vectors in $\mathcal{C}_K$ excluding the rows containing element in $\mathcal{I}$. **1** represents an all "1" column vector with the same number of entries as the row number of $\mathcal{C}_{K-1}$, and **2** represents an all "2" column vector with the same number of entries as the row number of $\mathcal{C}_{K-2}^{\backslash\{1\}}$, and so on. If $\mathcal{C}_{K-j}^{\backslash\{1,\ldots,j-1\}}$ for some j is empty, then the sub-matrix $[\ \mathbf{j} \quad \mathcal{C}_{K-j}^{\backslash\{1,\ldots,j-1\}}]$ will not appear in $\mathcal{C}_K$. In the above equations, each row vector contains q elements. For simplicity in representation, we omit the zeros at the end of each row. A full vector can be obtained by simply appending zeros at the end of each row to reach q elements.

Chapter 6

Exploring Dynamic and QIM Fingerprinting

In this chapter, we explore two directions for further study on multimedia fingerprinting, namely dynamic fingerprinting and QIM based multimedia fingerprinting with blind detection. The dynamic fingerprinting is designed for the applications allowing long term subscription from users, whereby users have access to multiple signals during a certain period. In this system, the fingerprinting scheme will change dynamically according to the observed collusion pattern to increase the probability of catching colluders. The second problem we consider in this chapter is the Quantization Index Modulation (QIM) fingerprinting scheme for blind detection. QIM is known for its excellent performance under blind detection, and thus we choose this embedding method in our study. Since there is no prior work in the literature on QIM based fingerprinting, we first explore how to apply QIM to the fingerprinting problem and then we will examine several QIM-based fingerprinting and compare their performance with spread-spectrum based fingerprinting.

6.1 Dynamic Fingerprinting for Multimedia

Nowadays, subscription based content services have become very popular, such as cable TV or online downloading, where users can obtain multimedia content from the content provider during the subscription period. One example is cable TV subscription service, through which people can have many options to view TV program or movies. Another example is the online subscription of movie download, where the user set up an account with the server, and download movies by signing in with his/her account. It is important to protect the content from unauthorized redistribution during the subscription period.

Most fingerprinting works address the anti-collusion problem for one signal in a static way, i.e. the fingerprint for each user is designed before-hand [70]. One of the first several works considering dynamic traitor tracing is by Fiat et al [23] and was improved in [6, 61]. In their work, the host signal is transmitted in segments, and the fingerprint in each segment is dynamically determined according to the detected fingerprints from previous segments. After collecting many segments, the detector makes a decision on the likely colluder. A major limitation of the work is the assumptions on real-time surveillance feedback and on dumb colluders, which may not always be realistic. Although it is possible to extend Fiat's dynamic fingerprinting to subscription scenarios of multiple programs by treating one movie as one segment, the detector has to collect tens of pirated movies for the algorithm to converge to catch ten colluders out of only 1000 users. If the total number of users scales up to millions, the detector has to collect nearly 100 pirated movies to catch colluders, which is impractical. Based on the work by Fiat et al., Safavi-Naini et al. proposed a sequential traitor tracing scheme [53], whereby the fingerprint for each signal segment is predetermined, rather than dynamic, and the detection is performed sequentially.

In this section, we consider the time dimension of the subscription based services and exploit the dynamics between the content owner and the colluders to design fingerprint [36]. Specifically, we adjust the fingerprint strength dynamically according to the colluders' information collected from previous pirated signals. To better understand the performance of the proposed scheme, we also examine possible strategies that colluders may take to combat dynamic fingerprinting. Results show that the proposed scheme has better collusion resistance than static ones and is robust to various collusion strategies.

6.1.1 Problem Description and Basic Fingerprinting Scheme

A dynamic fingerprinting scheme consists of several rounds. Each round i employs a basic fingerprinting system $\mathcal{F}_i$. The content owner distributes a signal $\mathbf{x}_i$ with fingerprint embedded to all users in the system. After receiving the fingerprinted signals, the colluders collectively generate a copy $\mathbf{z}_i$ and redistribute it. When a detector obtains a colluded copy $\mathbf{z}_i$, detection is performed on $\mathbf{z}_i$ to identify colluder(s). According to the detection results, the content owner redesign the fingerprint for round $i+1$ to increase the chances for colluders being caught. As a result, the fingerprinting scheme for round $i+1$, $\mathcal{F}_{i+1}$, is a function of $\mathcal{F}_i$ and the collusion strategy $\mathcal{K}_i$, i.e. $\mathcal{F}_{i+1} = f(\mathcal{F}_i, \mathcal{K}_i)$, where $f()$ is the dynamic fingerprinting strategy. The same process will continue in round $i+1$.

In this work, we employ orthogonal fingerprinting [71] as the basic fingerprinting scheme for each round and this basic fingerprinting scheme can be replaced by other fingerprinting systems [27, 28, 34] according to the application requirements. In orthogonal fingerprinting, mutually orthogonal spreading sequences $\{\mathbf{u}_j, j =$

$1...N_u\}$ with identical energy $||\mathbf{u}||^2$ are assigned to N_u users as the fingerprints. User j's fingerprinted copy is obtained as $\mathbf{y}_j = \mathbf{x} + \mathbf{u}_j$. After the distribution of the fingerprinted copies, the adversaries may employ various attacks $\mathcal{K}$. In this chapter we focus on averaging collusion [1], where colluders take the average of the corresponding signal in their copies to generate a colluded version. Additional distortion may be added to the multimedia signal during the collusion, which we model as an additive noise. Since few colluders would be willing to take higher risk than others, they generally would make contributions of approximately equal amount in the collusion. The colluded version $\mathbf{z}$ follows:

$$\mathbf{z} = \frac{1}{K} \sum_{j \in S_c} \mathbf{y}_j + \mathbf{d} = \frac{1}{K} \sum_{j \in S_c} \mathbf{u}_j + \mathbf{x} + \mathbf{d}, \tag{6.1}$$

where S_c is the colluder set with size K. The additional distortion is modelled as an *i.i.d.* additive Gaussian noise $\mathbf{d}$ with zero-mean and variance σ_d^2.

To identify colluders who have contributed to a suspicious copy of multimedia content, we employ a correlation detector commonly used for spread spectrum embedding. In this work, we focus on catching one colluder with high probability, for which the maximum detector [71] is employed. The user with the highest correlation with the test signal is identified as the colluder: $\hat{j} = arg \max_{j=1}^{N_u} T_j$, where

$$T_j = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{u}_j}{\sqrt{||\mathbf{u}||^2}} \quad j = 1, ..., N_u, \tag{6.2}$$

The colluder set in each round can be the same or different. In this chapter, we first consider the case of static colluder set, in which colluders remain the same for

[1] For orthogonal fingerprinting with Gaussian distributed fingerprints, a number of non-linear collusions employing order statistics, such as minimum collusion attack, have been shown [82] to be well approximated by the averaging collusion plus additive noise.

each round. In Section 6.1.3, we will discuss the dynamic strategies that colluders can employ in different rounds.

6.1.2 Dynamic Fingerprinting via Strength Scaling

Dynamic Fingerprinting Scheme

For simplicity, we start with a two-round system, and assumes the content owner initially does not have information about the colluders. In the first round, the content owner assumes every user has equal probability to collude, and the orthogonal fingerprinting with equal strength is employed. When a pirated copy is leaked, the correlation based detector in Eqn. (6.2) is employed to identify colluder. In this work, we choose not to immediately disconnect the colluder from the service. Notice that the action of disconnecting the detected colluder will inform the attackers that the content owner has identified some colluder(s), which will trigger complicated collusion strategies from the attackers. We will leave this case as our future work.

We design the fingerprints of the second round based on the detection statistic $T^{(1)}$ from the first round. Given the statistic $\{T_i^{(1)}\}$ for each user, a threshold h is chosen such that the users whose detection statistic is higher than h are put into the suspicious user set U_s. The fingerprint strength of the users in U_s will be increased by a small amount β in the second round to increase the probability of catching colluder(s). That is, for users in U_s, the fingerprinted copy is obtained as $\mathbf{y} = \mathbf{x} + (1+\beta)\mathbf{u}$. Other users' fingerprint strength remains as $\|\mathbf{u}\|$. The strategy of increasing only suspicious users' fingerprint energy instead of all the users is important in the applications where it is crucial to guarantee innocent users to get high-fidelity content. The parameters h and β in the proposed scheme enable the designer to achieve a trade-off between the detection performance and the received perceptual quality according to various applications' requirements.

After finding out a suspicious copy in the second round, the content owner employs the correlation detector in Eqn. (6.2) to identify colluder. As will be seen from Section 6.1.2, the increased fingerprint strength will increase the colluders' probability of being caught while the probability of false alarm remains unchanged. The detection statistic from second round $T_i^{(2)}$ can also be combined with $T_i^{(1)}$ as $T_i = (T_i^{(1)} + T_i^{(2)})/\sqrt{2}$ to facilitate the final decision, where we pick the user with highest T as the colluder.

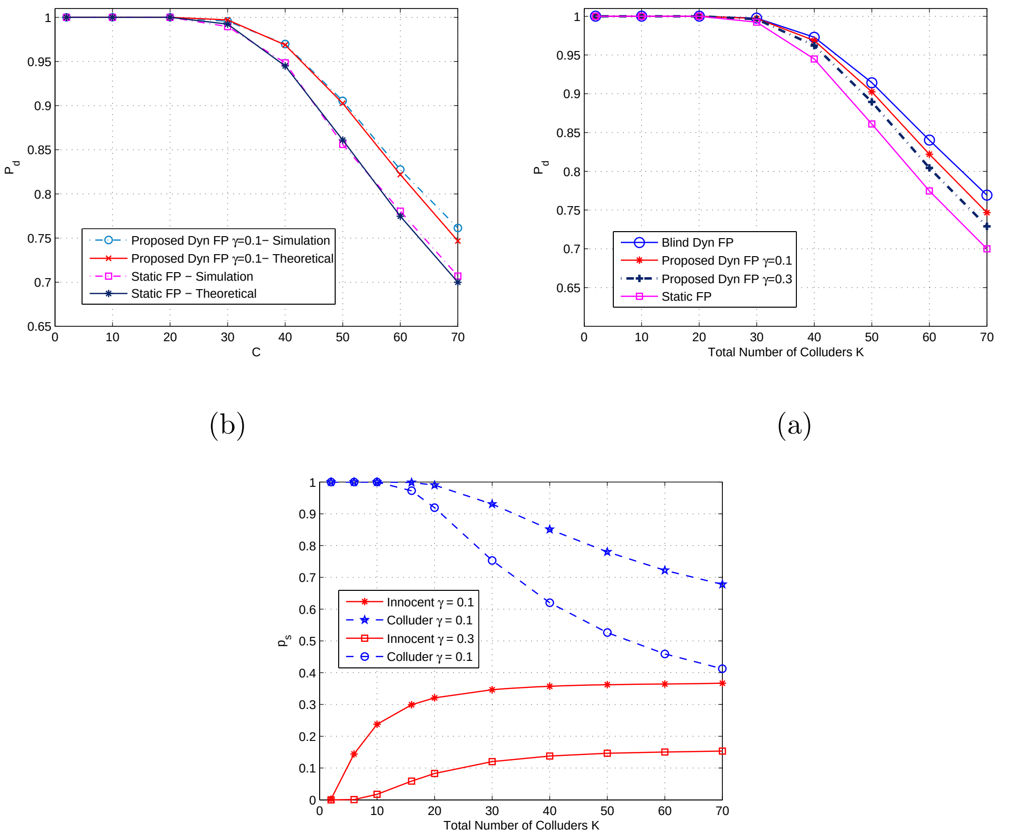
Analyzing Colluder Detection Performance

In this section, we analyze the probability of catching one colluder for the proposed dynamic fingerprinting. For comparison purpose, we also analyze two other alternatives: (1) repeatedly employing equal-energy orthogonal fingerprinting with the same fingerprint energy for both rounds, which we call *static fingerprinting*, and (2) employing equal-energy orthogonal fingerprinting for the first round and increasing the energy by β for all the users in the second round, called *blind dynamic fingerprinting* since it does not utilize the detection results from the first round.

For all three schemes, the first step is to determine the distribution of $T^{(1)}$ and $T^{(2)}$ so as to derive the distribution of the final detection statistic $T_i = (T_i^{(1)} + T_i^{(2)})/\sqrt{2}$ for each user. After obtaining the distribution of T_i's, we are able to calculate the probability of detection as

$$P_d = Pr(T_{M1} > T_{M2}) = \int Pr(T_{M1} > t) f_{T_{M2}}(t) dt \tag{6.3}$$

where $T_{M1} = \max_{i \in S_c} T_i, T_{M2} = \max_{i \notin S_c} T_i$, and $f_{T_{M2}}(t)$ is the $p.d.f.$ of T_{M2}. For all



(b)

(a)

(c)

Figure 6.1: Performance of three schemes: (a) Simulation results vs. analytical results; (b) Analytical results for all three schemes (c) Portion of users having higher fingerprint strength in the second round.

three schemes, T_i for innocent users are the same and follow Gaussian distributions $N \sim (0, \sigma_d^2)$. Thus we have

$$f_{T_{M2}}(t) = \frac{N_u - K}{\sigma_d} \Phi\left(\frac{t}{\sigma_d}\right)^{N_u-K-1} \times \phi\left(\frac{t}{\sigma_d}\right), \tag{6.4}$$

where N_u is the total number of users, and $\Phi()$ and $\phi()$ are the $c.d.f.$ and $p.d.f.$ of standard Gaussian distribution, respectively.

The detection statistic T for static fingerprinting and blind dynamic fingerprinting can be shown to follow

$$T_i \sim \begin{cases} N(0, \sigma_d^2), & i \notin S_c, \\ N(\mu_C, \sigma_d^2), & i \in S_c, \end{cases} \tag{6.5}$$

where μ_C takes value of $\sqrt{2}\|\mathbf{u}\|/K \triangleq \mu_{C1}$ for static fingerprinting and $(2 + \beta)\|\mathbf{u}\|/(\sqrt{2}K) \triangleq \mu_{C2}$ for blind dynamic fingerprinting.

To determine the distribution of T for the proposed dynamic fingerprinting, we need to first calculate the probability of $T_i^{(1)}$ having higher value than the threshold after the first round. This probability, denoted as p_{si}, is the probability for each user to be put into a suspicious user set. In this work, we set the threshold h adaptively as $h = \gamma \max_i T_i^{(1)}$. Then, the p_{si} is calculated as

$$p_{si} = \int \prod_{j \neq i} Pr(T_j^{(1)} < t/\gamma) f_{T_i}(t) dt. \tag{6.6}$$

Due to the equal energy and orthogonal fingerprint construction in the first round, p_{si} for all the innocent users are the same. We denote it as p_{s0}. Similarly, p_{si} for all the colluders would be the same under fair collusion, and we denote it as p_{s1}. We can show that T of the proposed scheme has the distribution of

$$T_i = \begin{cases} N(0, \sigma_d^2), & i \notin S_c, \\ N(\mu_{C2}, \sigma_d^2), & \text{with prob. } p_{s1} \quad i \in S_c \\ N(\mu_{C1}, \sigma_d^2), & \text{with prob. } 1 - p_{s1} \quad i \in S_c, \end{cases} \tag{6.7}$$

$$\text{and} \qquad Pr(T_{M1} > t) = 1 - \sum_{i=0}^{K} \Phi^i \left(\frac{t - \mu_{C2}}{\sigma_d} \right) \times \\ \Phi^{K-i} \left(\frac{t - \mu_{C1}}{\sigma_d} \right) \left(\begin{array}{c} K \\ i \end{array} \right) p_{s1}^i (1 - p_{s1})^{K-i}. \tag{6.8}$$

Plugging in the obtained results into Eqn. (6.3), we can obtain the probability of catching one colluder for all three schemes.

We validate the analysis through simulations with 5000 iterations. The examined system holds 1000 users, and the fingerprint length is 10^4, which is roughly the number of embeddable components in a 256×256 natural image. The first round of all three systems employs orthogonal fingerprinting with equal strength. In the second round, the static fingerprinting keeps the same fingerprint strength; the two dynamic fingerprinting schemes employ $\beta = 0.2$ to introduce a small amount of extra distortion, which is equivalent to 1.6dB loss in PSNR. Fig. 6.1(a) shows the simulation results on probability of detection P_d along with the numerical evaluation of Eqn. (6.3), where we select the results of static fingerprinting and the proposed dynamic fingerprinting as representatives. We can see that the analytical results match well with simulation results.

Comparison of Fingerprinting Schemes

In this section, we evaluate the performance of the proposed dynamic fingerprinting in comparison with the other two alternatives. The experimental settings are the same as above. The adaptive threshold γ for the proposed dynamic fingerprinting is set at 0.1 and 0.3, respectively. Fig. 6.1 (b) shows the analytical results of P_d versus the colluder number K at a Fingerprint-to-Noise Ratio of -5dB for all three schemes. We can see that the collusion resistance of the blind dynamic fingerprinting is better than that of static fingerprinting due to higher fingerprint strength in the second round. With the same detection probability, e.g. P_d=0.85, the static fingerprinting can only resist 51 colluders, while the blind dynamic fingerprinting can resist 59 colluders giving a 16% improvement in collusion resistance. The performance of the proposed dynamic fingerprinting lies in between and is close to blind dynamic fingerprinting scheme. For example, with $\gamma = 0.1$, the proposed dynamic fingerprinting can catch $6 \sim 7$ more colluders than static fingerprinting. As the threshold γ decreases, the collusion resistance of the proposed scheme gets closer to that of blind dynamic fingerprinting.

Although blind dynamic fingerprinting has the highest detection probability in all three schemes, everyone in the system, including innocent users, suffers a larger distortion in the second round of content distribution because of the increased fingerprint strength. This is unfair for the innocent users. In comparison, the proposed dynamic fingerprinting only increases the fingerprint strength for the suspicious users. Fig. 6.1(c) shows the portion of the innocent users' and colluders' fingerprint to be increased. From the results, we can see that with $\gamma = 0.1$, only 37% of the innocent users receive content with larger distortion in the second round and 68% of the colluders have their fingerprint energy increased, and we are able to achieve almost the same detection performance as the blind dynamic fingerprinting where all users have larger distortion. As we increase γ, fewer innocent users and colluders receive low quality signal, which leads a lower P_d. We can see that γ is a parameter used to achieve a trade-off between the collusion resistance performance and user satisfaction. Overall, the proposed dynamic fingerprinting has a better trade-off than the other two schemes.

6.1.3 Dynamic Collusion Strategies

The results shown in the last section are based on the assumptions that the colluders remain the same in both rounds. However, knowing the dynamic fingerprinting is employed, colluders may take different strategies in each round to circumvent the proposed fingerprinting. In this section, we examine the possible strategies that the colluders may take, and study the performance of the proposed dynamic

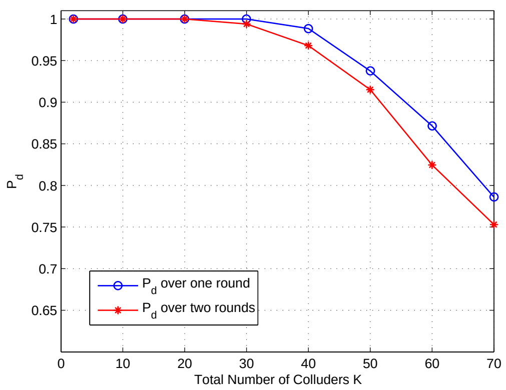


Figure 6.2: P_d of the proposed scheme against distinct colluder set.

fingerprinting against those collusion strategies. Here we assume that each of the colluders is honest to the coalition, and issues regarding selfish colluder can be studied following the framework in [81].

The effectiveness of the proposed scheme comes from the fact that the colluders participate the collusion in both rounds so that (1) the colluders may be detected as suspicious user in the first round and get fingerprint of increased strength for second round; (2) after participating the collusion in the second round, his/her probability of being detected is higher than before due to the increased fingerprint energy. Observing this, colluders may form different collusion sets for each round to circumvent the dynamic fingerprinting.

Suppose there are totally K colluders, denoted as S_c. The colluders decide to choose a subset of the colluders to collude in the first round and use a different subset of colluders for the second round. We denote the colluder set in the first round as S_{c1} with K_1 colluders, in the second round as S_{c2} with K_2 colluders, and $S_c = S_{c1} \cup S_{c2}$. The ratio of the colluders in the first round over the entire colluder group is denoted as $K_1/K = \eta$. We define an overlap ratio as $\xi = |S_{C1} \cap S_{C2}|/K$.

It is obvious that $K_2+K_1 = (1+\xi)K$. The parameters η and ξ feature the strategy employed by the colluders. For simplicity, we consider the colluder set for both rounds to have the same colluder number, which imposes one more constraint of $\eta = 1+\xi-\eta$. Under this model, the collusion strategy with repeated colluder set we have examined in Section 6.1.2 is a special case with $\eta = \xi = 1$. Now we examine other two cases, namely, disjoint colluder set and overlapped colluder set.

Disjoint Colluder Set In this strategy, the colluders divide themselves into two disjoint groups and each group performs collusion attack in one round, i.e. $\eta = 0.5$, $\xi = 0$. As a result, every colluder participates the collusion only in one round. Under this case, the detection statistic T based on both rounds has distribution close to that of the static fingerprinting as in Eqn. (6.5), and thus the detection performance would be similar to that of the static fingerprinting. However, due to the smaller colluder group in each round, the basic fingerprinting system in each round has a higher probability of detection as shown in Fig. 6.2. In this case, at each round, the detector is able to make decision without aggregating the detection statistics from multiple rounds, and the colluders actually have higher risk of being caught than before.

Overlapped Colluder Set In this case, some colluders only participate in one round of collusion and some participate in both rounds. The two parameters η and ξ are within the range of $0.5 < \eta < 1, 0 < \xi < 1$. In Fig. 6.3, we show the probability of detection under this collusion strategy, where we examined two settings: $\eta = 0.6$, $\xi = 0.2$ and $\eta = 0.9$, $\xi = 0.8$. Comparing the results with that of Section 6.1.2, we can see that the overlapped colluder set brings the detector up to 10% increase in probability of detection. As ξ and η approach 1, the performance approaches to the case with the same colluder set in both rounds. The strategy

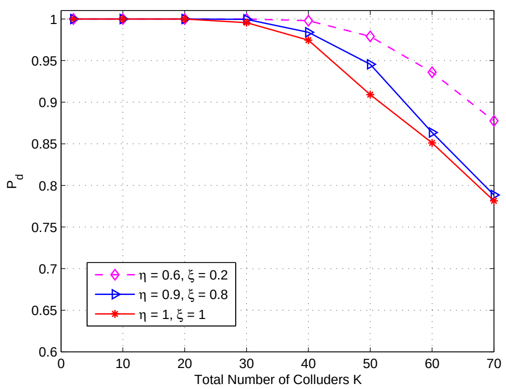


Figure 6.3: P_d of the proposed scheme against overlapped colluder set.

with overlapped colluder set does not bring benefit to the colluders. Fairness issues inside the colluders would also arise, which will be addressed in our future work.

In summary, if we look at only one round, both strategies with distinct and overlapped colluder set reduce the colluder number in each round and increase the colluders' risk of being caught; if we collectively examine two rounds, the overlapped strategy also increase the probability of detection. Therefore, the best strategy for colluders would be to try to collect as many colluders as possible in each round and launch the collusion attack altogether, which is the case that we have examined in Section 6.1.2.

6.2 QIM based Multimedia Fingerprinting

In the collusion-resistant multimedia fingerprinting literature, most of the schemes use spread spectrum techniques to embed the fingerprints. Under non-blind detection, spread spectrum based fingerprint embedding has high detection accuracy. However, in applications where the detection is performed without the original signal, the host signal acts as strong noise to the detector in the spread spectrum based fingerprinting and thus the detection accuracy is very low. An important alternative to spread spectrum embedding is Quantization Index Modulation (QIM) [11, 12]. In QIM, the host data is quantized using multiple quantizers, the index of which is chosen based on the message to be embedded. The advantages of the QIM embedding is the high detection accuracy under blind detection. In this section, we explore the possibility of employing QIM for anti-collusion fingerprinting applications. Specifically, we employ dither modulation (DM) for the fingerprint embedding [11]. We have observed that the existing DM algorithm primarily focusses on embedding binary bits. We first construct a basic embedding scheme to resist collusion attacks by extending the existing DM to embed multiple symbols, and study its performance. To better understand the results, we introduce a general theoretical model and analyze the collusion resistance of DM based fingerprinting. From our theoretical analysis, we infer that fingerprint sequences with low correlation have better collusion resistance. We then design a new algorithm to construct dither sequences so that the resulting fingerprints have low correlation and are approximately orthogonal. We demonstrate through simulations that our proposed method performs better than the basic scheme, and compare the results with those obtained using spread spectrum based fingerprinting. Our results show that the fingerprint correlation is not easy to control through QIM embedding and hence it does not perform as well as the spread spectrum based fingerprinting even under non-blind detection.

Spread Transform Dithered Modulation (STDM) is an alternative robust quantization based embedding approach whereby a random unitary transformation is applied to the signals before quantization. With this embedding method, every bit of information can be spread over the signal. This would have similar effect as spread spectrum based embedding, and the quantization operation during the embedding would bring benefits in blind detection scenario. Meanwhile, we notice that the existing QIM embedding techniques are well defined for embedding binary bits, and our results on DM based fingerprinting show that it is non-trivial to extend these methods to embed non-binary symbols. In principle, it would be possible to construct binary fingerprint sequences employing collusion-secure codes such as Boneh-Shaw's [9] for fingerprinting multimedia. However, since these codes are designed without considering the embedding issues explicitly, they are often too long to be reliably embedded [9], and/or are unable to resist a nontrivial number of colluders [5]. For example, to attain moderate levels of collusion resistance such as to resist 10 colluders out of 1000 users, the Boneh-Shaw code requires a long codeword at least on the order of 10^6 bits. Such high payloads often exceed the embedding capacity for most multimedia data under stringent robustness requirements. In this section, we propose to use non-binary fingerprint code, such as traceability code employed in [33] and map each symbol to a binary codeword through an efficient construction for embedding [59].

6.2.1 Fingerprinting Model and QIM Review

Spread Spectrum based Fingerprinting

Spread spectrum embedding has been widely used for multimedia fingerprinting [33,64,71]. One typical example is orthogonal fingerprinting, whereby mutually orthogonal spreading sequences are generated as fingerprint for each user. Another way to construct fingerprint is to employ a coding step, such as error correcting code (ECC), and map symbols in the alphabet to orthogonal sequences [30]. The

i^{th} user's fingerprinted copy $\mathbf{y}_i$ is obtained by adding his/her fingerprint sequence $\mathbf{s}_i$ to the host signal $\mathbf{x}$, i.e.

$$\mathbf{y}_i = \mathbf{x} + \mathbf{s}_i. \tag{6.9}$$

After the fingerprinted copies reach end users, some users may mount collusion attacks and try to remove the traces of the embedded fingerprint. Averaging collusion plus additive noise is mostly studied in the literature [71, 76] and a number of non-linear collusions have been shown to be well approximated by this model [82]. Under averaging collusion, the resulting signal, $\mathbf{z}$, is the average of c colluders' fingerprinted copy:

$$\mathbf{z} = \frac{1}{c} \sum_{i \in S_c} \mathbf{y}_i + \mathbf{n}, \tag{6.10}$$

where S_c is the colluder set containing c colluders, $\mathbf{n} = [n_1, n_2, \ldots, n_N]^T$ is additive noise that models additional distortions applied on the colluded signal, and N is the length of the fingerprint sequence. For simplicity, we assume $\mathbf{n}$ follows an *i.i.d.* Gaussian distribution.

The goal of the detector is to catch at least one of the colluders with a high probability given the suspicious copy, $\mathbf{z}$. As the host signal can be made available to detectors in many fingerprinting applications, we subtract the host signal from the suspicious copy to obtain a test signal. Match filter detector is then employed to find the colluder; that is, we correlate the test signal with each of the N_u spreading sequences (one for each user) and identify the sequence that gives the maximum correlation. The detection statistic for the i^{th} user is defined as

$$T_i = \frac{(\mathbf{z} - \mathbf{x})^T \mathbf{s}_i}{\sqrt{||\mathbf{s}_i||^2}}, \tag{6.11}$$

and the $\hat{m}^{th}$ user is declared as a colluder if

$$\hat{m} = \arg\max_{i=1,2,\ldots,N_u} T_i. \tag{6.12}$$

Quantization Index Modulation(QIM)

Dither Modulation(DM) In quantization based methods, the host data is quantized using multiple quantizers and the index of the quantizer is chosen based on the message to be embedded [11]. A simple way to build multiple quantizers is by dither modulation (DM). Specifically, for a host-signal $\mathbf{x}$, the embedding function for hiding binary messages can be written as

$$\mathbf{q}_{x_i} = Q_{\Delta}(\mathbf{x} + \mathbf{d}_i) - \mathbf{d}_i \quad \forall i \in \{0, 1\}, \tag{6.13}$$

where $Q_{\Delta}(.)$ represents the quantization function with step size Δ and $\mathbf{d}_i$ represents the dither sequence that is used to perturb the host signal before quantization. One possible way to construct the dither sequence is by first choosing one dither vector (say $\mathbf{d}_0$) as *i.i.d.* random variables following a uniform distribution over $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ and then the second one can be obtained using [11]

$$d_{1k} = \begin{cases} d_{0k} + \frac{\Delta_k}{2} & \text{if} \quad d_{0k} < 0, \\ d_{0k} - \frac{\Delta_k}{2} & \text{if} \quad d_{0k} \geq 0, \end{cases} \qquad \forall k \in \{1, 2, \dots, N\}, \tag{6.14}$$

where $\mathbf{d}_i = [d_{i1}, d_{i2}, \dots, d_{iN}]^T$.

It has been shown in [11, 20] that the rate-distortion and robustness tradeoff can be improved in the basic QIM method by compensation and other postprocessing operations. In the distortion compensated QIM (DC-QIM), a fraction of the quantization error is added back to the original signal. Thus, the watermarked image can be represented as

$$\mathbf{y}_i = \alpha \left(Q_{\frac{\Delta}{\alpha}}(\mathbf{x} + \mathbf{d}_i) - \mathbf{d}_i \right) + (1 - \alpha)\mathbf{x} \quad \forall i \in \{0, 1\}, \tag{6.15}$$

where the constant α can be chosen appropriately to maximize the Signal-to-Noise Ratio (SNR) [11] or to maximize embedding capacity [20, 45].

Spread Transform Dither Modulation (STDM) Another robust way to implement QIM is STDM. Instead of applying scalar DM directly on each component of host signal, STDM first applies a random unitary transformation by projecting the host signal $\mathbf{x}$ onto a random direction, $\mathbf{u}$. The projection values, $x^{(p)} = \mathbf{u}^T\mathbf{x}$, are then quantized using DM to obtain the watermarked signal [11], i.e.

$$\begin{aligned} \mathbf{y} &= \mathbf{x} + (y^{(p)} - x^{(p)})\mathbf{u}, && (6.16) \\ y^{(p)} &= Q_{\triangle}(x^{(p)} + d_b) - d_b, \quad b \in \{0,1\}, && (6.17) \end{aligned}$$

where b is the message bit to be embedded. Typically, the projection direction $\mathbf{u}$ is randomly generated according to a secret key, and therefore we do not need to introduce uncertainty in the choice of dither sequence d_b. In our implementation, we choose the dither sequences to be deterministic. Due to random projections, only the noise in the direction of $\mathbf{u}$ would affect performance. Thus, the STDM provides a higher effective Watermark to Noise Ratio (WNR), and is more robust against additive noise attacks [11].

During the detection, the test signal $\mathbf{z} = \mathbf{y} + \mathbf{n}$, is projected onto vector $\mathbf{u}$ to get $z^{(p)} = \mathbf{z}^T\mathbf{u}$. The embedded bit is determined as

$$\hat{m} = \begin{cases} \arg\min_{b=0,1} \|z^{(p)} - (Q_{\triangle}(x^{(p)} + d_b) - d_b)\| & \text{for non-blind detection,} \\ \arg\min_{b=0,1} \|z^{(p)} - (Q_{\triangle}(z^{(p)} + d_b) - d_b)\| & \text{for blind detection.} \end{cases} \tag{6.18}$$

6.2.2 Dither Modulation based Fingerprinting

Extending QIM to Fingerprinting

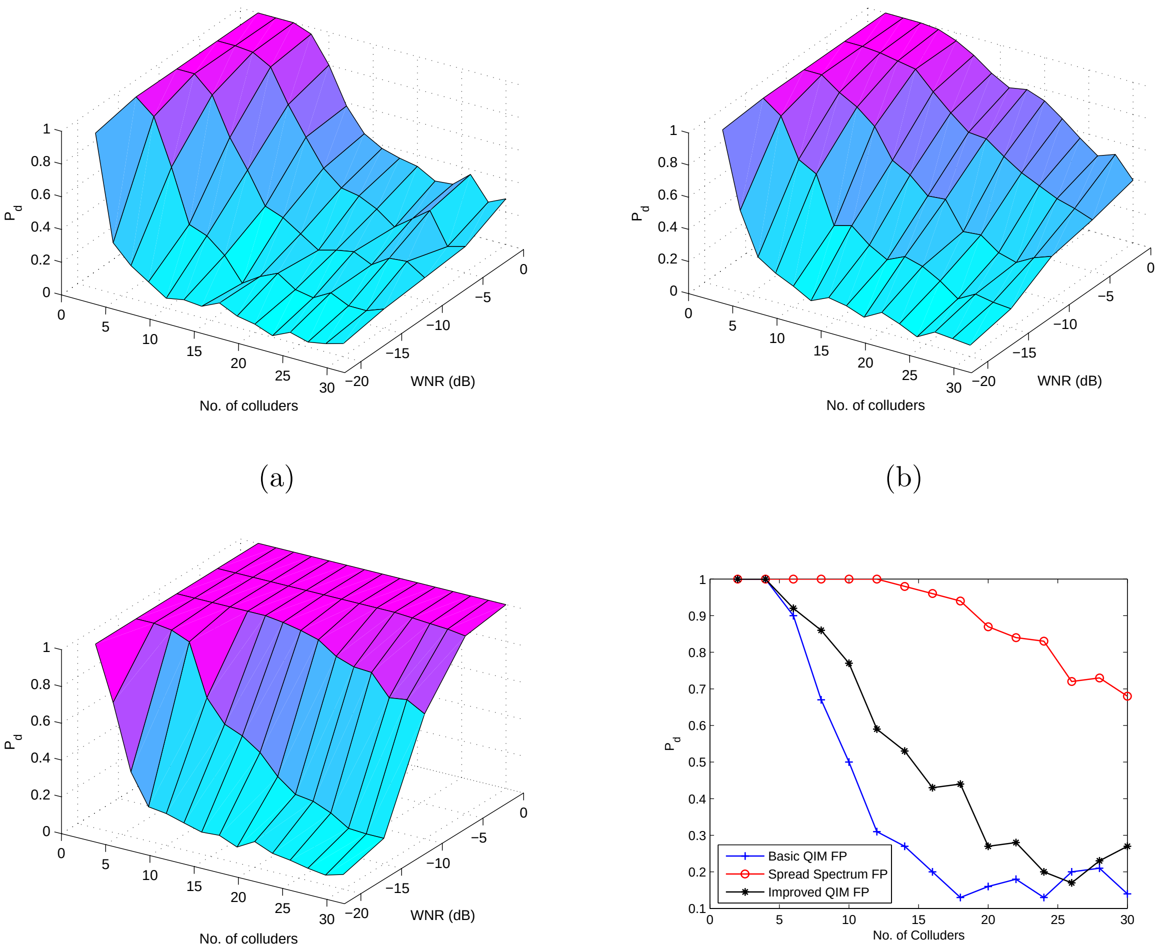
It is known in the recent literature that lattice-based quantizers can be used to embed multiple alphabets [80], but they generally have a very high computational complexity. To overcome this problem, we consider a simple extension of the DM scheme for embedding multiple symbols, i.e. use mutually orthogonal dither sequences for each user. Specifically, we construct N_u random dither sequences $\mathbf{d}_i$ following an *i.i.d.* Gaussian distribution such that $E(\mathbf{d}_i^T\mathbf{d}_j) = 0\ \forall i, j \in \{1, ..., N_u\}$ and $i \neq j$. The fingerprinted copies are then obtained using Eqn. (6.15).

When the content owner obtains the suspicious copy $\mathbf{z}$, he/she can apply maximum likelihood detection, which would involve an exhaustive search over $O(2^{N_u})$ different colluder combinations. Although this detector is optimal in minimizing the probability of detection error, its complexity is very high and grows exponentially with the number of users. Therefore, in our implementation, we apply the minimum-distance detection as used in the QIM literature [11] to find one of the colluders. More specifically, the $\hat{m}^{th}$ user is declared a colluder if

$$\hat{m} \quad = \quad \arg\min_{k=1,2,...N_u} ||\mathbf{z} - \mathbf{y}_k||^2. \tag{6.19}$$

This detector also provides a fair comparison with the spread spectrum based fingerprinting employing match filter detection of Eqn. (6.11).

We simulated this basic scheme for $N_u = 1024$ users under averaging collusion on a 256×256 size Lena image with the PSNR of the fingerprinted image with respect to the original set to 42dB. The embedding was done in the block DCT domain and the quantization step sizes were chosen according to the JPEG quantization table. We examine the probability of catching one colluder, P_d, at different watermark-to-noise-ratio (WNR), and the results are shown in Fig. 6.4(a). For comparison purposes, we show in Fig. 6.4(c) the performance of a spread spectrum based fingerprinting under the same conditions. From the results, we observe that the basic QIM-based fingerprinting can only resist about half dozen colluders at moderate to high WNRs, while spread spectrum based fingerprinting can resist



(a) (b)

(c) (d)

Figure 6.4: Comparison on the performance of QIM-based and spread spectrum based fingerprinting: (a) Basic QIM-based Fingerprinting; (b) Improved QIM-based Fingerprinting; (c) Spread spectrum based Fingerprinting; (d) Results under WNR= −10dB.

more than 30 colluders with high probability in the same WNR range. To facilitate the analysis of results, we build a theoretical model to study the detection performance in the next subsection.

Theoretical Analysis of QIM-based Fingerprinting

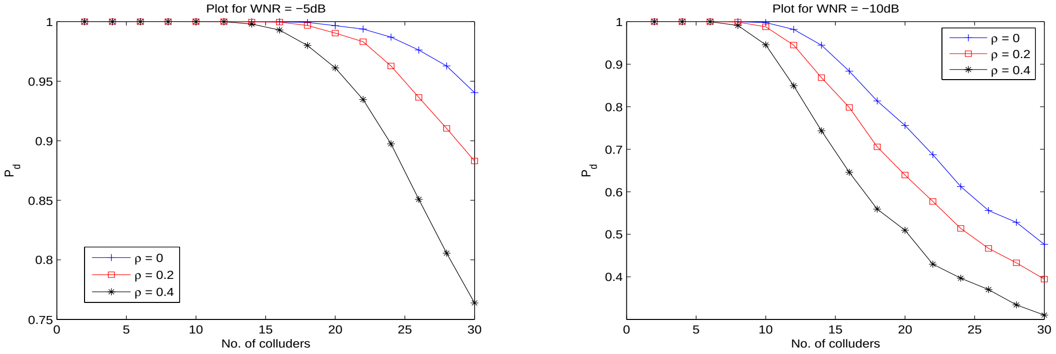
Without loss of generality, we assume the first c colluders perform averaging collusion as formulated in Eqn. (6.10). Then, the probability of catching one colluder, P_d, is

$$P_d = Pr(\min(X_1, X_2, \ldots, X_c) < \min(X_{c+1}, X_{c+2}, \ldots, X_{N_u})), \tag{6.20}$$

where $X_k = ||\mathbf{z} - \mathbf{y}_k||^2$ is the detection statistic for user k. We can show that for a system with totally N_u users and c colluders, $\mathbf{X} = [X_1, X_2, \ldots, X_{N_u}]^T$ approximately follows a multi-variate Gaussian distribution with mean and covariance matrix given by:

$$\begin{aligned} m_k = E(X_k) &= \begin{cases} \left(\frac{c-1}{c}\right)\frac{\Lambda}{2} + N\sigma_n^2 & \text{if} \quad 1 \le k \le c, \\ \left(\frac{c+1}{c}\right)\frac{\Lambda}{2} + N\sigma_n^2 & \text{if} \quad c+1 \le k \le N_u, \end{cases} && (6.21) \\ R(i,j) = cov(X_i, X_j) &= 2\sigma_n^4 \left[N + \left(\frac{\Lambda}{\sigma_n^2}\right) \frac{P(i,j)}{c} \right], && (6.22) \end{aligned}$$

where N is the length of fingerprint sequence, σ_n^2 is variance of the additive noise, and Λ is the average mean square difference between two fingerprinted copies. The



(a) WNR = −5dB

(b) WNR = −10dB

Figure 6.5: Theoretical results on probability of correct detection with respect to number of colluders for different ρ values

matrix $P_{(N_u \times N_u)}$ is given by

$$P(i,j) = \begin{cases} c-1 & \text{if } 1 \leq i,j \leq c \quad \text{and } i = j, \\ -1 & \text{if } 1 \leq i,j \leq c \quad \text{and } i \neq j, \\ 0 & \text{if } 1 \leq i \leq c \quad \text{and } j > c, \\ 0 & \text{if } 1 \leq j \leq c \quad \text{and } i > c, \\ c+1 & \text{if } c < i,j \leq N_u \quad \text{and } i = j, \\ 1 & \text{if } c < i,j \leq N_u \quad \text{and } i \neq j. \end{cases} \tag{6.23}$$

The detailed derivation is described in Section 6.2.4. We remark that the above theoretical framework is general and applicable to any fingerprinting scheme as long as the distance between any pair of fingerprints is identical. Thus, this model can help explain the results obtained by spread spectrum techniques as well.

From Eqn. (6.21), we notice that the difference between the means of X_k for the colluders and the innocent users is $\Delta m = \frac{\Lambda}{c}$. Thus, we infer that the average performance in terms of probability of catch one colluder would improve as the distance between two fingerprint sequences, Λ, is increased, or the number of colluders c is decreased. This result can also be interpreted in terms of the correlation between the fingerprint sequences $\rho = 1 - \frac{\Lambda}{2W}$ (W is the average energy of the fingerprint). In Fig. 6.5, we show the probability of correct decision P_d for different values of the correlation parameter ρ by numerically evaluating the theoretical model. We observe from the plot that the performance of the fingerprinting system increases when ρ reduces (or Λ increases). Based on this principle, we examine the correlation for basic QIM-based fingerprinting. We observe that the main reason for our basic QIM construction not performing well compared to the spread spectrum case is because the resulting correlation value $\rho = 0.45$ was much higher than that of the spread spectrum based fingerprinting (close to zero). Therefore, in order to improve the collusion resistance of QIM-based fingerprinting, we need to carefully select dither sequences so that the resulting fingerprint sequences have low correlation. In the next section, we propose a new technique that will help reduce the correlation and improve the detection performance.

Improved Dither Sequence Construction for QIM-based Fingerprinting

According to the theoretical model, for best results, the dither sequences should be constructed so as to make the final fingerprints have as low correlation as possible. The problem can be formulated as

$$\min \quad (Q_\Delta(\mathbf{x}+\mathbf{d}_i)-\mathbf{x}-\mathbf{d}_i)^T(Q_\Delta(\mathbf{x}+\mathbf{d}_j)-\mathbf{x}-\mathbf{d}_j), \quad \forall\, i,j \in \{1,2,\ldots,N_u\}, i \neq j, \tag{6.24}$$

subject to the fairness constraints that the fingerprint energies for different users are equal, i.e.

$$(Q_\Delta(\mathbf{x}+\mathbf{d}_i)-\mathbf{x}-\mathbf{d}_i)^T(Q_\Delta(\mathbf{x}+\mathbf{d}_i)-\mathbf{x}-\mathbf{d}_i) = W, \quad \forall i = 1,2,\ldots,N_u. \tag{6.25}$$

Let $\Delta = [\Delta_1, \Delta_2, \ldots, \Delta_N]^T$, where Δ_k is the step size of the uniform quantizer in the k^{th} component. We can show that the quantization operation (for the mid-raiser quantizer) is given by

$$Q_\Delta(\mathbf{x}+\mathbf{d}_i) = \mathbf{a} + \frac{1}{2}\Delta \otimes \mathbf{Y}_i, \tag{6.26}$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_N]^T$, $\mathbf{Y}_i = [Y_{i1}, Y_{i2}, \ldots, Y_{iN}]^T$ and $\Delta \otimes \mathbf{Y}_i = [\Delta_1 Y_{i1}, \Delta_2 Y_{i2}, \ldots, \Delta_N Y_{iN}]^T$. The corresponding k^{th} element in the vector can be represented as

$$a_k = \begin{cases} t_k\Delta_k & \text{if} \quad t_k\Delta_k \leq x_k < (t_k+0.5)\Delta_k, \\ (t_k+1)\Delta_k & \text{if} \quad (t_k+0.5)\Delta_k \leq x_k < (t_k+1)\Delta_k; \end{cases} \tag{6.27}$$

$$Y_{ik} = \begin{cases} -1 & \text{if} \quad \frac{-\Delta_k}{2} \leq d_{ik} < (a_k - x_k), \\ 1 & \text{if} \quad (a_k - x_k) \leq d_{ik} < \frac{\Delta_k}{2}. \end{cases} \tag{6.28}$$

Here, we assume that $-\frac{\Delta_k}{2} \leq d_{ik} < \frac{\Delta_k}{2}$. Note that a_k is a multiple of the quantization step size Δ_k, that is closest to the host data sample x_k. Further, the value of a_k is independent of the choice of the dither sequence. The term $\frac{1}{2}\Delta_k Y_{ik}$ denotes the residue term that would choose one among the two nearby quantization points based on the value of the dither sequence.

By substituting Eqns. (6.27) and (6.28) back into the minimization problem, and using the Lagrange multipliers to incorporate the equal-energy constraints we obtain an equivalent cost function$-J$ given by

$$\begin{aligned} J &= (\mathbf{a}-\mathbf{x}+\frac{1}{2}\Delta\otimes\mathbf{Y}_i-\mathbf{d}_i)^T(\mathbf{a}-\mathbf{x}+\frac{1}{2}\Delta\otimes\mathbf{Y}_j-\mathbf{d}_j) \\ &+ \nu_1\left((\mathbf{a}-\mathbf{x}+\frac{1}{2}\Delta\otimes\mathbf{Y}_i-\mathbf{d}_i)^T(\mathbf{a}-\mathbf{x}+\frac{1}{2}\Delta\otimes\mathbf{Y}_i-\mathbf{d}_i)-W\right) \\ &+ \nu_2\left((\mathbf{a}-\mathbf{x}+\Delta\otimes\mathbf{Y}_j-\mathbf{d}_j)^T(\mathbf{a}-\mathbf{x}+\frac{1}{2}\Delta\otimes\mathbf{Y}_j-\mathbf{d}_j)-W\right), \end{aligned} \tag{6.29}$$

where ν_1 and ν_2 are Lagrange multiplier constants. Setting the gradient of J with respect to both the dither vectors to zero, we get a set of linear equations solving which we obtain

$$\mathbf{d}_i = \frac{1}{2}\Delta \otimes \mathbf{Y}_i - \kappa_i(\mathbf{x} - \mathbf{a}), \tag{6.30}$$

where κ_i are scalars chosen so that the total energy of the fingerprint is equal to W. In our implementations, we first choose the vectors $\mathbf{Y}_i \in \{-1, 1\}^N$ for each user. The dither sequences are then generated according to Eqn. (6.30). In the next section, we present the results for this scheme and compare it with our basic dither based approach presented earlier in Section 6.2.2.

Results and Discussions

To examine the effectiveness of the proposed improvement algorithm, we apply the constructed dither sequences on the Lena image with the same parameter settings as in Section 6.2.2; that is, 256 × 256 Lena image fingerprinted with a PSNR of 42dB and $N_u = 1024$ users. The results are shown in Fig. 6.4(b) alongside the corresponding plots for the basic QIM scheme and spread spectrum fingerprinting. For better illustration, we compare the performance of the three schemes at WNR $= -10$dB in Fig. 6.4(d). We observe that the improved scheme performs much better than the basic scheme. This gain can be attributed to the reduced average correlation among fingerprint sequences in the improved scheme (around 0.1), compared to a high value of 0.45 in the basic scheme. We also observe that our improved scheme still does not perform as well as the traditional spread spectrum based scheme. A closer examination shows that the variance of the correlation statistic for QIM based fingerprinting is larger (ρ values range from -0.15 to 0.2), while the spread spectrum based fingerprinting has correlation ranging from -0.04 to 0.04. Owing to the nonlinear quantization operation employed in QIM, it is not easy to control the correlation between the fingerprints for a total of N_u users as in spread spectrum based fingerprinting.

From the results, we can see that in the proposed DM based fingerprinting it is non-trivial to construct dither sequences to get fingerprint sequences with low correlation. Since the QIM has been well studied for embedding binary bits, a natural way of using QIM for fingerprinting is to embed binary fingerprint codeword. In the mean time, we observe that STDM based embedding has an effect of spreading the embedded bit over the host signal. By choosing mutually orthogonal projection vectors for different bits, we can achieve an effect similar to the overlapped spread spectrum embedding. Taking these two factors into consideration, we explore the STDM based coded fingerprinting in the next section.

6.2.3 STDM based ECC fingerprinting

ECC based fingerprinting with spread spectrum embedding has been shown very promising in providing an excellent trade-off between the collusion resistance and detection efficiency [34]. In this section, we explore the performance of STDM based ECC fingerprinting. For embedding, we propose to map each symbol to a binary codeword that is constructed to well separate q symbols.

Fingerprint Embedding and Detection

To embed a q-ary fingerprint codeword with length L_1, we partition the host signal into L_1 segments. In each segment, we choose a simplex code $S(L_2, m, D)$ to represent each of the q symbols. A simplex code of dimension m has $q = 2^m$ codewords, each of length $L_2 = 2^m - 1$ and provides an equal distance of $D = 2^{m-1}$. Simplex code has good properties such as a large relative distance (> 0.5) and a non-trivial code rate; and thus it can support a large alphabet size q with better separation among symbols. The binary simplex codeword for each symbol is embedded into a segment of the host signal through STDM, where we project the host signal to L_2 mutually orthogonal random directions and quantize the resulting projection values. This has an overall effect of overlapped embedding that the bits representing one symbol are added on top of each other and spread over the segment. An illustration is shown in Fig. 6.6.

Table 6.1: Quantization Error for Different Bit Positions (Qstep = 4)

Bit Position	1	2	3	4	5	6	7	8	9	...
Bit 0	1.744	3.388	3.987	1.718	0.970	2.334	0.810	0.796	0.0098	...
Bit 1	2.256	0.612	0.013	2.282	3.030	1.666	3.190	3.204	3.9902	...

During our preliminary exploration, we found that due to the randomness of the projection, the quantization error, or the energy for bit 0 and 1 is not equal. Table 6.1 list the quantization error for each bit position. We can see that in some positions, bit 0 is embedded with more energy than bit 1, and vice versa for other places. This unevenness would make the collusion resistance unpredictable and make the fingerprinted signals have different visual quality. One way to overcome this problem is to adjust the dither sequence according to the host signal to make bit 0 and 1 have equal embedding energy. However, under blind detection, the detector would not be able to identify the correct dither sequences from the received signal, which will incur decoding error. In this chapter, we propose to spread each bit of the simplex codeword to multiple bits by mapping bit 1 to a l-bit random binary sequence $\mathbf{s}$ and 0 to $\bar{\mathbf{s}}$, the bit-wise flipped version of $\mathbf{s}$. The spreading factor l can be adjusted to tradeoff the perceptibility and robustness. For clear presentation, we shall use "logic bit" to refer to the bit in the simplex codeword, and "bit" refers to the bit in the spreading sequence $\mathbf{s}$. Every bit in the sequence

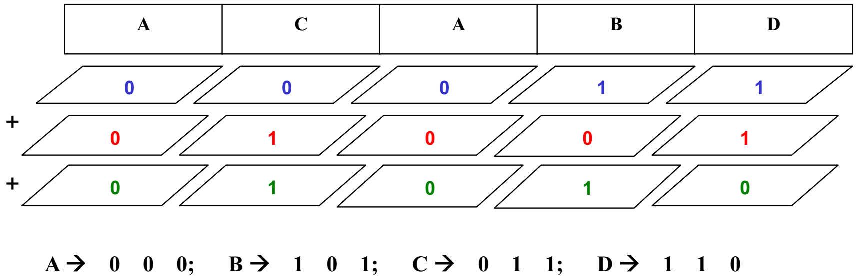


Figure 6.6: An example of embedding q-ary codeword using STDM.

$\mathbf{s}$ is embedded into the same segment using STDM by projecting the signal onto a random direction. As a result, a total of $L_2 l$ bits for all the L_2 logic bit in a simplex codeword are superposed and spread over one segment of host signal. The k^{th} fingerprinted segment $\mathbf{y^{(k)}}$ can be represented as

$$\mathbf{y^{(k)}} = \mathbf{x^{(k)}} + \sum_{i=1}^{L_2} \sum_{j=1}^{l} (y_{ij}^{(p)} - x_{ij}^{(p)}) \mathbf{u_{ij}}, \tag{6.31}$$

where $\mathbf{u_{ij}}$ is the projecting direction for the i^{th} bit in logic bit j's spreading sequence; $x_{ij}^{(p)}$ is the projection of the k^{th} segment host signal $\mathbf{x^{(k)}}$ on $\mathbf{u_{ij}}$, and $y_{ij}^{(p)}$ is obtain by quantizing $x_{ij}^{(p)}$ using Eqn. (6.16).

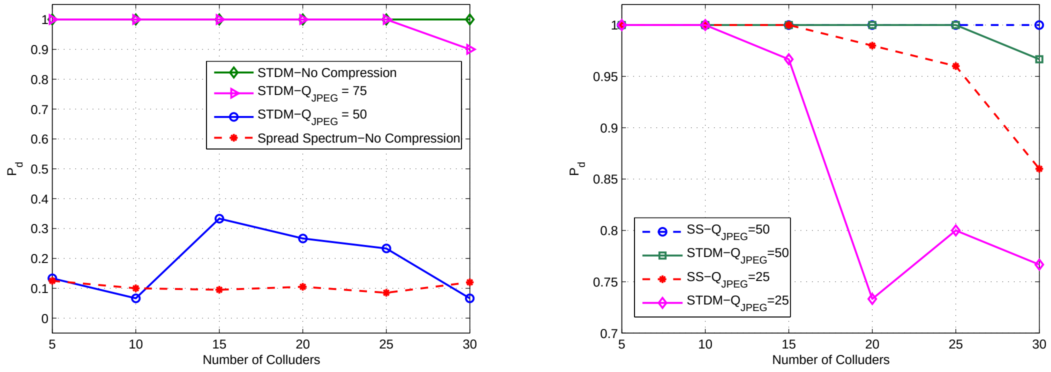
During the detection, we first calculate the distance information for each bit according to Eqn. (6.18). Then we add all these distance information from every bit of each user's fingerprint codeword. The user who has the smallest distance with the test signal is declared as colluder.

Results and Discussions

We test the performance of the proposed STDM based ECC fingerprinting on a 256×256 Lena image. We choose Reed-Solomon code (14, 2, 13) as the fingerprint code with code length 14, dimension 2, and alphabet size 16. Each of the 16 symbols is mapped to a binary codeword of a simplex code (15, 4, 8). Mutually orthogonal spreading sequences are chosen for projecting the input data and the resulting values are quantized using the binary dither modulation method. As mentioned earlier, we choose deterministic dither sequences $d_0 = 0$ and $d_1 = \triangle/2$ to maximize separation. The PSNR of the fingerprinted copy is set at 40.8 dB.

In Fig. 6.7, we show the probability of catching one colluder, P_d, under averaging collusion and additional JPEG compression. The results for the blind and non-blind scenarios are shown in Fig. 6.7(a) and (b) respectively. We notice that under moderate JPEG compression, the system is able to resist at least a few dozen users' collusion in both cases. When the JPEG quality factor reduces, the performance drops sharply in the case of blind detection even for a small number of colluders. This is expected because the projected point $z^{(p)}$ moves outside the correct decoding region when a large JPEG quantization step size is used. This leads to wrong estimates of the true projection points $x^{(p)}$, eventually resulting in a large probability of decoding error. On the other hand, in the case of non-blind detection, the projected point $z^{(p)}$ provides some information for correct decoding. Therefore, the performance of non-blind detection degrades gracefully as the JPEG quality factor reduces and the number of colluders increases.

To facilitate comparison, we also implement the spread spectrum based ECC fingerprinting with the same Reed-Solomon code, i.e. each symbol is mapped to an orthogonal spreading sequences before embedding [34]. Matched filter detection is employed for catching one colluder. Also in Fig. 6.7, we show the results for spread spectrum based fingerprinting under both the blind detection and non-blind detection. Under blind detection, we notice that the STDM based fingerprinting has significant advantage over spread spectrum based fingerprinting with up to 90% increase in P_d. On the other hand, the spread spectrum based ECC fingerprint-



(a) (b)

Figure 6.7: Simulation results of STDM based and spread spectrum based ECC fingerprinting under averaging collusion and JPEG compression: (a) blind detection; (b) non-blind detection.

ing performs a little better than STDM based scheme under non-blind detection with less than 5% increase on P_d under moderate JPEG compression. This is because the spread spectrum based scheme employs orthogonal modulation to embed each symbol, while STDM based scheme use a simplex code, which does not perform as well as orthogonal modulation in separating different symbols. Overall, the proposed STDM based fingerprinting shows significant advantages over spread spectrum based fingerprinting under blind detection and slightly reduced performance under non-blind detection.

6.2.4 Appendix: Theoretical Model on QIM based Fingerprinting

In this appendix, we present the theoretical analysis on the performance of fingerprinting schemes employing minimum distance decoding. Let $\mathbf{x}$ denote the host signal and $\mathbf{y}_i$ represent user i's fingerprinted copy. In the case of QIM, $\mathbf{y}_i$

is obtained by quantizing the host signal $\mathbf{x}$ as shown in Eqn. (6.15). Under the averaging collusion model, the received signal, $\mathbf{z}$, is given by

$$\mathbf{z} = \frac{1}{c}\sum_{i=1}^{c} \mathbf{y}_i + \mathbf{n}, \tag{6.32}$$

where $\mathbf{n}$ denotes the additive noise used to model any further processing. Here, we assume, without loss of generality, that the first c colluders participate in collusion. The decoder applies minimum distance decoding as given in Eqn. (6.19) to find one of the colluders.

The probability of catching one colluder, P_d, is given by

$$P_d = Pr(\min(X_1, X_2, \ldots, X_c) < \min(X_{c+1}, X_{c+2}, \ldots, X_{N_u})), \tag{6.33}$$

where $X_k = ||\mathbf{z} - \mathbf{y}_k||^2$ is the detection statistic for user k. Substituting for $\mathbf{z}$ from Eqn. (6.32), we get

$$X_k = ||\mathbf{n} + \frac{\alpha}{c}\sum_{i=1}^{c}\left(\mathbf{q}_{x_i} - \mathbf{q}_{x_k}\right)||^2. \tag{6.34}$$

The detection statistic X_k is a random variable and its distribution would depend on the noise statistics. The mean of X_k can be obtained as

$$m_k = E(X_k) = \mathbf{s}_k^T\mathbf{s}_k + trace(\Sigma_n), \tag{6.35}$$

where Σ_n is the covariance matrix of the noise variable $\mathbf{n}$ and $\mathbf{s}_k = \frac{\alpha}{c}\sum_{i=1}^{c}\left(\mathbf{q}_{x_i} - \mathbf{q}_{x_k}\right)$ gives the average difference between the quantization points among the users in collusion set. In a similar note, the $(i,j)^{th}$ element of the covariance matrix R of the detection statistics X_k can be expressed as

$$R(i,j) = cov(X_i, X_j) = \mathbf{w}_n^{(4)^T}\mathbf{1}_N - trace(\Sigma_n^T\Sigma_n) + 2\mathbf{w}^{(3)^T}(\mathbf{s}_i + \mathbf{s}_j) + 4\mathbf{s}_i^T\Sigma_n\mathbf{s}_j, \tag{6.36}$$

where $\mathbf{1}_N$ denotes a column vector with all N elements as 1 and $\mathbf{w}_n^{(l)}$ is a $N \times 1$ vector in which the i^{th} element represents the l^{th} order moment of the corresponding noise component n_i.

If we assume that the noise $\mathbf{n}$ is Gaussian with zero mean and variance σ_n^2, then the detection statistic would follow the chi-square distribution [47] and its mean and variance can be simplified as

$$\begin{aligned} m_k &= \mathbf{s}_k^T \mathbf{s}_k + N\sigma_n^2, & (6.37) \\ R(i,j) &= 2N\sigma_n^4 + 4\sigma_n^2 \mathbf{s}_i^T \mathbf{s}_j. & (6.38) \end{aligned}$$

We remark that as the length of the fingerprint N is increased, the detection statistic can be well approximated as a multi-variate Gaussian distribution [47]. Substituting for $\mathbf{s}_i$, we obtain

$$\mathbf{s}_i^T \mathbf{s}_j = \left(\frac{\alpha}{c}\right)^2 \sum_{l=1}^{c} \sum_{k=1}^{c} (\mathbf{q}_{x_l} - \mathbf{q}_{x_i})^T (\mathbf{q}_{x_k} - \mathbf{q}_{x_j}),$$

which can be further reduced to give

$$\mathbf{s}_i^T \mathbf{s}_j = \frac{P(i,j)\Lambda}{2c}. \qquad (6.39)$$

Here $P(i,j)$ is given as in Eqn. (6.23) and Λ is the average mean squared difference between any two fingerprinted copies,

$$\Lambda = \frac{2}{N_u(N_u-1)} \sum_{i=1}^{N_u} \sum_{j=1, j\neq i}^{N_u} ||\mathbf{y}_i - \mathbf{y}_j||^2. \qquad (6.40)$$

Substituting for $\mathbf{s}_i^T \mathbf{s}_j$ from Eqn. (6.39) into equations (6.37) and (6.38), we obtain the desired expressions as given in equations (6.21) and (6.22).

6.3 Chapter Summary

In this chapter, we have studied two problems related to collusion-resistance multimedia fingerprinting. The first one is anti-collusion fingerprinting for applications with long-term subscription, where a group of pirates may launch several rounds of collusions. We propose a dynamic fingerprinting strategy to adjust the fingerprint strength in each round according to the detection results from previous round. Both analytical and simulation results show that the proposed scheme performs better, in terms of detection probability, than static fingerprinting and close to blind dynamic fingerprinting without having as many users suffering from reduced visual quality. Dynamic collusion strategies are also examined, where the results indicate that the best strategy for colluders is to gather as many colluders as possible in each round of the collusion.

The second problem we have considered is to use Quantization Index Modulation (QIM) embedding for fingerprinting applications mainly for blind detection scenario. In particular, we use spread transform dither modulation (STDM) to embed the fingerprint code, where each q-ary symbol is mapped to a binary simplex code for embedding. The results show significant advantage of STDM based embedding over spread spectrum based embedding under blind detection, where the STDM based fingerprinting has up to 90% improvement in probability of detection over spread spectrum based fingerprinting. This suggests that STDM is a promising embedding technique for fingerprinting under blind detection scenarios.

Chapter 7

Conclusions and Future Perspectives

In this dissertation, we have studied various aspects of fingerprint design in collusion-resistant fingerprinting for multimedia signals.

Starting from a cross-layer framework for multimedia fingerprinting, we first examine the end-to-end performance of ECC-based fingerprinting by considering both the coding and embedding layers. The results show that traditional ECC-based fingerprinting has high efficiency in distribution of fingerprinted signal and colluder detection but rather limited collusion resistance. The ECC based fingerprinting motivates us to find avenues to improve its collusion resistance while preserving its efficient detection and distribution.

Based on the observation from the performance examination, we propose two new 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 retain 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.

Building upon the proposed joint coding and embedding framework, we consider fingerprinting video signal under such challenging settings as to accommodate millions of users and to resist hundreds of users' collusion. We further address issues of designing code structure and speeding up detection. Our proposed trimming detection approach can speed up the detection by 3 orders of magnitude with only slightly drop of detection accuracy. With the proposed fingerprint construction and efficient detection, the system holding 16 million users can resist 50-60 colluders' interleaving collusion and more than 100 users' averaging collusion as well as 80 users' non-linear collusion. The user capacity and collusion resistance can be further increased by adjusting such system parameters as k and β. Both the analysis and the experimental results show a strong potential of joint coding and embedding ECC fingerprinting for large-scale video fingerprinting applications.

The increase in the popularity of the subscription based services, such as cable TV, motives us to study the problem of anti-collusion fingerprinting for applications with long-term subscription, where a group of pirates may launch several rounds of collusions. We propose a dynamic fingerprinting strategy to adjust the fingerprint strength in each round according to the detection results from previous rounds. Both analytical and simulation results show that the proposed scheme performs better, in terms of detection probability, than static fingerprinting and close to blind dynamic fingerprinting without having as many users suffering from reduced visual quality. Dynamic collusion strategies are also examined, where the results indicate that the best strategy for colluders is to gather as many colluders as possible in each round of the collusion.

We notice that spread spectrum embedding has been widely used in the existing multimedia fingerprinting schemes because the host signal is often available at the detector and thus can facilitate the detection. However, in applications where the host signal is not easy to obtain, the spread spectrum based fingerprinting would have low detection accuracy. In this dissertation, we have explored a class of quantization based embedding methods for fingerprinting applications, whose advantage is the high detection accuracy under blind detection. Specifically, we explore coded fingerprinting based on spread transform dither modulation (STDM) embedding. Simulation results show that this coded STDM based fingerprinting has significant advantages over spread spectrum based fingerprinting under blind detection, where the STDM based fingerprinting has up to 90% improvement in probability of detection over spread spectrum based fingerprinting.

Based on the study of this dissertation, there are several aspects of multimedia fingerprinting that can be further explored. First, our current joint-coding-embedding fingerprinting is mainly examined on uncompressed video data. The effect of video compression is treated as an additional distortion to our fingerprint detection. In some scenario, the raw video data may not be available to the fingerprint embedder. For example, in cable TV application, the set-up boxes, where the fingerprint is embedded, have very limited computing and storage resources and cannot afford to do the embedding in fully-uncompressed domain. It is desirable to embed fingerprint in the compressed domain of the signal to reduce the computation and delay incurred by the fingerprint embedding. Apparently, results from the existing work on the uncompressed domain cannot be directly applied to this case due to different nature of compressed and uncompressed data [67, 68]. Thus we need to consider the particular property of the embedding domain and jointly explore the coding and embedding layers for fingerprint design.

Second, this dissertation initiates the research on collusion-resistant dynamic fingerprinting for multimedia and as the first step of exploration, proposes a simple but quite effective scheme. There still remain many research questions. From designer's perspective, we are interested in the possible strategies to dynamically design the fingerprints. The choice of the strategy will be closely related to the collusion pattern observed from the previous rounds. Then how to accurately estimate the collusion pattern and design effective fingerprinting given the estimated collusion pattern would become two important questions that need to be answered first. On the other hand, the adversaries will try to seek for the best strategy to minimize their risk of being caught given the designer's dynamic strategy. Thus the interactions between fingerprint designer and attackers is also an interesting topic to explore, where game theory [25, 46] can be used as a tool to model and solve the problem.

Finally, the fingerprinting research in this dissertation is to prevent each individual user from sharing his/her content with others. A new paradigm for next-generation multimedia content protection may encourage and take advantage of file sharing among users [10,42]. This would open up opportunities of new technology-economics model that builds an incentive-based off-line market for digital media, and supports the legitimate resale by individual users to others such that both consumers and copyright owners/service provider can benefit from the resale revenues. In this new system, the fingerprinting would play an important role to provide content protection after the system being hacked. Moreover, the embedded fingerprints can also serve as a track mark to facilitate the study of the viral structure, which is an important input information for the operation of the new system. It would be interesting to investigate how to design fingerprints so that it can play such roles in the new model.

BIBLIOGRAPHY

[1] http://www.dmwmedia.com/news/2006/05/03/wsj-new-mpaa-study-puts-studio-piracy-losses-75-higher.

[2] http://www.cnn.com/2004/world/asiapcf/03/17/predator.video/.

[3] Sample video sequences from Technische Universitat Munchen, Germany: http://www.ldv.ei.tum.de/page70?lang=en.

[4] N. P. Anthapadmanabhan, A. Barg, and I. Dumer. Fingerprinting Capacity Under the Marking Assumption. *submitted to IEEE Transactions on Information Theory*, December 2006, available at http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0612073.

[5] A. Barg, G.R. Blakley, and G. Kabatiansky. Digital Fingerprinting Codes: Problem Statements, Constructions, Identification of Traitors. *IEEE Trans. on Information Theory*, 49(4):852–865, April 2003.

[6] O. Berkman, M. Parnas, and J. Sgall. Efficient Dynamic Traitor Tracing. *SIAM Journal of Computing*, 30(6):1802–1828, 2001.

[7] G. R. Blakley, C. Meadows, and G. B. Purdy. Fingerprinting Long Forgiving Messages. In *Lecture Notes in Computer Sciences*, volume 218, pages 180–189, June 1986.

[8] D. Boneh and J. Shaw. Collusion-secure Fingerprinting for Digital Data. In *Proc. Crypto, LNCS*, volume 963, pages 452–465, 1995.

[9] D. Boneh and J. Shaw. Collusion-secure Fingerprinting for Digital Data. *IEEE Trans. on Information Theory*, 44(5):1897–1905, September 1998.

[10] R. Cattelan, S. He, and D. Kirovski. Prototyping a Novel Platform for Free-Trade of Digital Content. In *WebMedia*, 2006.

[11] B. Chen and G. W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Trans. on Information Theory*, 47(4):1423–1443, May 2001.

[12] B. Chen and G. W. Wornell. Quantization Index Modulation Methods for Digital Watermarking and Information Embedding of Multimedia. *The Journal of VLSI Signal Processing*, 27(1-2):7–33, November 2004.

[13] B. Chor, A. Fiat, and M. Naor. Tracing Traitors. In *Crypto'94, Lecture Notes in Computer Science*, volume 839, pages 257–270, 1994.

[14] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing Traitors. *IEEE Trans. on Information Theory*, 46(3):893–910, May 2000.

[15] H. Chu, L. Qiao, and K. Nahrstedt. A Secure Multicast Protocol with Copyright Protection. In *ACM SIGCOMM Computer Communications Review*, volume 32, pages 42–60, April 2002.

[16] I. Cox, J. Bloom, and M. Miller. *Digital Watermarking: Principles and Practice*. Morgan Kaufmann, 2001.

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

[18] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Images, Audo and Video. In *IEEE International Conference on Image Processing*, volume 3, pages 243–246, September 1996.

[19] H. A. David. *Order Statistics*. New York: John Wiley and Son, 2nd Edition, 1981.

[20] J. J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod. Scalar Costa Scheme for Information Embedding. *IEEE Trans. on Information Theory*, 51(4):1003–1019, April 2003.

[21] F. Ergun, J. Kilian, and R. Kumar. A Note on the Limits of Collusion-Resistant Watermarks. In *Lecture Notes in Computer Sciences*, volume 1592, January 1999.

[22] M. Fernandez and M. Soriano. Soft-Decision Tracing in Fingerprinted Multimedia Content. *IEEE Multimedia*, 11(2):38–46, April-June 2004.

[23] A. Fiat and T. Tassa. Dynamic Traitor Tracing. *Journal of Cryptography*, 14(3):211–223, 2001.

[24] J. Fridrich. Visual Hash for Oblivious Watermarking. In *SPIE Conf. Security, Stegonography and Watermarking of Multimedia Content*, volume 3971, pages 286–294, January 2000.

[25] D. Fudenberg and J. Tirole. *Game Theory.* The MIT Press, Cambridge, Massachusetts, 1991.

[26] V. Guruswami and M. Sudan. Improved Decoding of Reed-Solomon and Algebraic-geometry Codes. *IEEE Trans. on Information Theory*, 45(6):1757–1767, September 1999.

[27] O. Harmanci and M. K. Mihcak. Complexity-Regularized Video Watermarking via Quantization of Pseudo-Random Semi-Global Linear Statistics. In *European Signal Processing Conference(EUSIPCO)*, 2005.

[28] O. Harmanci and M. K. Mihcak. Motion Picture Watermarking via Quantization of Pseudo-Random Linear Statistics. In *Visual Communications and Image Processing Conference*, 2005.

[29] F. Hartung, J. Su, and B. Girod. Spread Spectrum Watermarking: Malicious Attacks and Counterattacks. In *SPIE Conf. Security, Stegonography and Watermarking of Multimedia Content*, pages 147–158, January 1999.

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

[31] S. He and M. Wu. Group-Oriented Joint Coding and Embedding Technique for Multimedia Fingerprinting. In *SPIE Conf. Security, Stegonography and Watermarking of Multimedia Content*, volume 5681, pages 96–105, January 2005.

[32] S. He and M. Wu. Improving Collusion Resistance of Error Correcting Code Based Multimedia Fingerprinting. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1029–1032, March 2005.

[33] S. He and M. Wu. Performance Study on Multimedia Fingerprinting Employing Traceability Codes. In *International Workshop on Digital Watermarking (IWDW)*, Sept. 2005.

[34] S. He and M. Wu. Joint Coding and Embedding Techniques for Multimedia Fingerprinting. *IEEE Trans. on Information Forensics and Security*, 1(2):231–247, June 2006.

[35] S. He and M. Wu. Collusion-Resistant Video Fingerprinting for Large User Group. In *IEEE International Conference on Image Processing*, October 2006.

[36] S. He and M. Wu. Collusion-Resistant Dynamic Fingerprinting for Multimedia. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2007.

[37] S. He and M. Wu. Collusion-Resistant Video Fingerprinting for Large User Group. *IEEE Trans. on Information Forensics and Security*, accepted, June 2007.

[38] S. He and M. Wu. Adaptive Detection for Group-based Multimedia Fingerprinting. *IEEE Signal Processing Letters*, accepted, June 2007.

[39] H. Jin and J. Lotspiech. Attacks and Forensic Analysis for Multimedia Content Protection. In *IEEE International Conference on Multimedia and Expo*, pages 1392–1395, July 2005.

[40] H. Jin and J. Lotspiech. Hybrid Traitor Tracing. In *IEEE International Conference on Multimedia and Expo*, pages 1329–1332, July 2006.

[41] H. Jin, J. Lotspiech, and S. Nusser. Traitor Tracing for Prerecorded and Recordable Media. In *Proc. of ACM Workshop on Digital Rights Management*, pages 83–90, October 2004.

[42] D. Kirovski and K. Jain. Off-line Economies for Digital Media. In *ACM International Workshop on Network and Operating Systems Support for Digital Audio and Vidoe*, 2006.

[43] D. Kirovski and H.S. Malvar. Spread-spectrum Watermarking of Audio Signals. *IEEE Trans. on Signal Processing*, 51(4):1020–1033, April 2003.

[44] R. Koetter and A. Vardy. Algebraic Soft-Decision Decoding of Reed-Solomon Codes. *IEEE Trans. on Information Theory*, 49(11):2809–2825, November 2003.

[45] P. Moulin and R. Koetter. Data-Hiding Codes. *Proceedings of IEEE*, 93(12):2083–2126, December 2005.

[46] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* The MIT Press, Cambridge, Massachusetts, 1994.

[47] A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes.* McGraw Hill Publications, 2002.

[48] S. Paul. *Multicast on the Internet and Its Application.* Kluwer Academic Publisher, 1998.

[49] C. Podilchuk and W. Zeng. Image Adaptive Watermarking using Visual Models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–540, May 1998.

[50] H. V. Poor. *An Introduction to Signal Detection and Estimation.* Springer Verlag, 1999.

[51] J. G. Proakis. *Digital Communications.* McGraw-Hill, 2001.

[52] R. Safavi-Naini and Y. Wang. Collusion Secure q-ary Fingerprinting for Perceptual Content. In *Security and Privacy in Digital Rights Management (SP-DRM'01)*, pages 57–75, 2002.

[53] R. Safavi-Naini and Y. Wang. Sequential Traitor Tracing. *IEEE Trans. on Information Theory*, 49(5):1319–1326, May 2003.

[54] R. Safavi-Naini and Y. Wang. Traitor Tracing for Shortened and Corrupted Fingerprints. In *Proc. of Digital Right Management (DRM'02)*, pages 81–100, 2003.

[55] A. Silverberg, J. Staddon, and J. L.Walker. Applications of List Decoding to Tracing Traitors. *IEEE Trans. on Information Theory*, 49(5):1312–1318, May 2003.

[56] J.N. Staddon, D. R. Stinson, and R. Wei. Combinatorial Properties of Frameproof and Traceability Codes. *IEEE Trans. on Information Theory*, 47(3):1042–1049, March 2001.

[57] K. Su, D. Kundur, and D. Hatzinakos. Spatially Localized Image-dependent Watermarking for Statistical Invisibility and Collusion Resistance. *IEEE Trans. on Multimedia*, 7(1):52–56, February 2005.

[58] K. Su, D. Kundur, and D. Hatzinakos. Statistical Invisibility for Collusion-resistant Digital Video Watermarking. *IEEE Trans. on Multimedia*, 7(1):43–51, February 2005.

[59] A. Swaminathan, S. He, and M. Wu. Exploring QIM based Anti-Collusion Fingerprinting for Multimedia. In *SPIE Conf. Security, Stegonography and Watermarking of Multimedia Content*, volume 6072, January 2006.

[60] M. D. Swanson, B. Zhu, and A. H. Tewfik. Multiresolution Scene-based Video Watermarking Using Perceptual Models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, May 1998.

[61] T. Tassa. Low Bandwidth Dynamic Traitor Tracing Schemes. *Journal of Cryptography*, 18(2):167–183, 2005.

[62] D. To, R. Safavi-Naini, and Y. Wang. A 2-secure Code with Efficient Tracing Algorithm. In *Lecture Notes in Computer Science*, volume 2551, pages 149–162, 2002.

[63] Y. L. Tong. *The Multivariate Normal Distribution.* Springer-Verlag, 1990.

[64] W. Trappe, M. Wu, Z.J. Wang, and K.J.R. Liu. Anti-collusion Fingerprinting for Multimedia. *IEEE Trans. on Signal Processing*, 51(4):1069–1087, April 2003.

[65] T. van Trung and S. Martirosyan. New Constructions for IPP Codes. In *IEEE International Symposium on Information Theory*, 2003.

[66] T. van Trung and S Martirosyan. On a Class of Traceability Codes. *Designs, Codes and Cryptography*, 31(2):125–132, February 2004.

[67] A. Varna, S. He, A. Swaminathan, and M. Wu. Analysis of Non-linear Collusion Attacks on Fingerprinting Systems for Compressed Multimedia. In *IEEE International Conference on Image Processing*, to appear September 2007.

[68] A. Varna, S. He, A. Swaminathan, M. Wu, H. Lu, and Z. Lu. Collusion-Resistant Fingerprinting for Compressed Multimedia Signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2007.

[69] N.R. Wagner. Fingerprinting. In *Proceedings of the 1983 Symposium on Security and Privacy*, pages 18–22, April 1983.

[70] Z.J. Wang, W. Trappe M. Wu, and K.J.R. Liu. Group-Oriented Fingerprinting for Multimedia Forensics. *EURASIP Journal on Applied Signal Processing, Special Issue on Multimedia Security and Rights Management*, 2004(14):2153–2173, October 2004.

[71] 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 Processing*, 14(6):804–821, June 2005.

[72] S. B. Wicker. *Error Control Systems for Digital Communication and Storage.* Prentice Hall, 1995.

[73] M. Wu and B. Liu. Data Hiding in Image and Video: Part-I – Fundamental Issues and Solutions. *IEEE Trans. on Image Processing*, 12(6):685–695, June 2003.

[74] M. Wu and B. Liu. *Multimedia Data Hiding.* Springer Verlag, 2003.

[75] M. Wu and B. Liu. Data Hiding in Binary Image for Authentication and Annotation. *IEEE Trans. on Multimedia*, 6(4):528–538, August 2004.

[76] M. Wu, W. Trappe, Z. Wang, and K.J.R. Liu. Collusion Resistant Fingerprinting for Multimedia. *IEEE Signal Processing Magazine, Special Issue on Digital Rights Management*, 21(2):15–27, March 2004.

[77] M. Wu, H. Yu, and B. Liu. Data Hiding in Image and Video: Part-II – Designs and Applications. *IEEE Trans. on Image Processing*, 12(6):696–705, June 2003.

[78] Y. Yacobi. Improved Boneh-Shaw Content Fingerprinting. In *CT-RSA 2001, Lecture Notes in Computer Sciences*, volume 2020, pages 378–391, 2001.

[79] F. Zane. Efficient Watermark Detection and Collusion Security. In *FC 2000, Lecture Notes in Computer Science*, volume 1962, pages 21–32, 2001.

[80] Q. Zhang and N. Boston. Quantization Index Modulation using the e_8 Lattice. In *Proceedings of the 41th Annual Allerton Conference on Communication, Control and Computing*, October 2003.

[81] H. V. Zhao and K. J. R. Liu. Risk minimization in traitors within traitors in multimedia forensics. In *IEEE International Conference on Image Processing*, September 2005.

[82] H. V. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu. Forensic Analysis of Nonlinear Collusion Attacks for Multimedia Fingerprinting. *IEEE Trans. on Image Processing*, 14(5):646–661.