COLLUSION-RESISTANT FINGERPRINTING FOR MULTIMEDIA IN A BROADCAST CHANNEL ENVIRONMENT

A Thesis

by

WILLIAM LUH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Electrical Engineering

COLLUSION-RESISTANT FINGERPRINTING FOR MULTIMEDIA IN A BROADCAST CHANNEL ENVIRONMENT

A Thesis

by

WILLIAM LUH

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Deepa Kundur
(Chair of Committee)

Don R. Halverson
(Member)

Aniruddha Datta
(Member)

Riccardo Bettati
(Member)

Chanan Singh
(Head of Department)

December 2004

Major Subject: Electrical Engineering

ABSTRACT

Collusion-Resistant Fingerprinting for Multimedia
in a Broadcast Channel Environment. (December 2004)

William Luh, B.A., University of Toronto
Chair of Advisory Committee: Dr. Deepa Kundur

Digital fingerprinting is a method by which a copyright owner can uniquely embed a buyer-dependent, inconspicuous serial number (representing the fingerprint) into every copy of digital data that is legally sold. The buyer of a legal copy is then deterred from distributing further copies, because the unique fingerprint can be used to trace back the origin of the piracy. The major challenge in fingerprinting is collusion, an attack in which a coalition of pirates compare several of their uniquely fingerprinted copies for the purpose of detecting and removing the fingerprints.

The objectives of this work are two-fold. First, we investigate the need for robustness against *large* coalitions of pirates by introducing the concept of a *malicious distributor* that has been overlooked in prior work. A novel fingerprinting code that has superior *codeword length* in comparison to existing work under this novel malicious distributor scenario is developed. In addition, ideas presented in the proposed fingerprinting design can easily be applied to existing fingerprinting schemes, making them more robust to collusion attacks.

Second, a new framework termed *Joint Source Fingerprinting* that integrates the processes of watermarking and codebook design is introduced. The need for this new paradigm is motivated by the fact that existing fingerprinting methods result in a perceptually undistorted multimedia after collusion is applied. In contrast, the new paradigm equates the process of collusion amongst a coalition of pirates, to degrading the perceptual characteristics, and hence commercial value of the multimedia in question. Thus by enforcing that the process of collusion diminishes the commercial value of the content, the pirates are deterred from attacking the fingerprints. A fingerprinting algorithm for video as well as an efficient means of broadcasting or distributing fingerprinted video is also presented. Simulation results are provided to verify our theoretical and empirical observations.

ACKNOWLEDGMENTS

I would like to sincerely thank my thesis advisor, Dr. Deepa Kundur, for her invaluable advice, strong support, patience and confidence in me as a researcher. Without Dr. Kundur, this thesis would not have been possible.

I would also like to thank members of my research committee, Dr. Riccardo Bettati, Dr. Aniruddha Datta, and Dr. Don R. Halverson, for their patience and response to any questions or needs that I have passed their way.

I would like to give special thanks to Alexandra Czarlinska for her caring friendship, advice and strong support. In addition, I would like to thank my friends and colleagues, Udit Budhia, Anli Chen, Raghav Dube, Carlos J. Moreira, and Unoma Okorafor, for their support and advice. Last, but not least, I would like to thank my mother, father, and sister, for their support and confidence that I can do what my heart is set to do.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

The ease at which digital data can be perfectly reproduced has made piracy, the illegal distribution of content, a growing threat for content distributors and copyright holders. As illegal copies of digital data, such as video, and audio proliferate over the Internet, an emerging interest in protecting copyrighted material has surfaced. One such method of protecting copyrighted material is called *digital fingerprinting.* Digital fingerprinting is a method by which a copyright owner can uniquely embed a buyer-dependent, inconspicuous serial number (representing the fingerprint) into every copy of digital data that is legally sold. The buyer of a legal copy is then deterred from distributing further copies, because the unique fingerprint can be used to trace back the origin of the piracy. In this sense, fingerprinting is a *passive* form of security effective after an attack has been applied, which is in contrast to *active* forms of security, such as encryption, that prevent the attack in the first place.

The major challenge in fingerprinting is that all legally distributed copies of the same digital data are *similar*, with the exception of the unique buyer-dependent fingerprints. A coalition of pirates who possess distinctly fingerprinted copies of the same data can therefore exploit this diversity, by comparing their digital data, and possibly detecting, and then rendering the fingerprints unreadable. Such an attack is known as *collusion.* One goal of fingerprinting is thus to ensure that some part of the fingerprint is capable of surviving a collusion attack, so as to identify at least one of the pirates.

For multimedia fingerprinting, an extra level of robustness is required compared

The journal model is *IEEE Transactions on Automatic Control.*

to other types of data. Robustness is equivalent to the fingerprint's ability to remain traceable after intentional or unintentional modification of the fingerprinted media. In general, multimedia can withstand some amount of (single) user-generated distortion, such that this distortion is imperceptible to the end-user. A coalition of pirates might then individually modify their multimedia in addition to applying the collusion attack. Hence a fingerprinting scheme for multimedia should also be robust to some amount of user-generated distortion. Examples of common user-generated distortions are additive white Gaussian noise, linear filtering such as blurring with Gaussian or Laplacian point spread functions, JPEG compression, geometric distortions such as cropping and resizing, among others [1, 2]. Since fingerprinting has the goal of traceability, fingerprinting for digital media should be robust to both collusion as well as user-generated distortions.

In applications such as *Video on Demand* (VoD), it is impractical to send a unique fingerprinted video to each subscriber, because bandwidth usage is excessive. The solution to this problem is to send identical digital data to all subscribers, and then at the user end, build a uniquely fingerprinted video. Such a scheme is referred to as *fingerprinting in a broadcast channel environment*.

A. Contributions of This Thesis

The contributions of this thesis are two-fold. First, a novel fingerprinting code that has superior *codeword length* (in comparison to existing work under assumptions such as the "malicious distributor" to be described later) is developed. Existing fingerprinting schemes strive to achieve codes that are minimal in codeword length, yet possess a tolerable probability of error in the face of collusion [3, 4, 5, 6, 7]. This thesis motivates the need for robustness against large coalition sizes (i.e., large numbers of colluders), by introducing the concept of a malicious distributor that has been overlooked in prior work. The proposed fingerprinting code is shown to be more robust than existing codes for such a problem. Finally, ideas presented in the proposed fingerprinting design can easily be applied to existing fingerprinting schemes, making them more robust to collusion attacks.

The second major contribution is the introduction of a new framework that integrates the processes of watermarking and codebook design (called the traditional *multi-step paradigm* in this thesis). This novel single-step approach is termed *Joint Source Fingerprinting* (JSF). The JSF approach is fundamentally different because it equates the process of collusion amongst a coalition of pirates, to degrading the perceptual characteristics, and hence commercial value of the multimedia in question. Thus by enforcing that the process of collusion diminishes the commercial value of the content, the colluders are deterred from attacking the fingerprints. This characteristic is in direct contrast to existing fingerprinting methods that result in a perceptually identical multimedia after collusion is applied. In addition to proposing the new framework, a working algorithm for digital video and simulation results are provided to verify empirical and theoretical observations.

B. Structure of This Thesis

This section gives the reader an overview of the chapters to follow. Chapter II, *Problem Formulation*, presents the traditional fingerprinting problem for multimedia, showing how general codes, modulation, watermarking, and broadcasting processes are integrated to produce a fingerprinted media that is distributed to an end-user. The problem addressed in this thesis is limited in scope to the novel design of the fingerprinting codebook. Attacks on fingerprinting systems are then presented, introducing the reader to three particular types of attacks: codebook, single-user, and multimedia collusion. The chapter concludes by defining the exact problem studied in this thesis.

The purpose of Chapter III, entitled *Literature Review*, is to survey research pertaining primarily to codebook design, the focus of this work. The chapter briefly introduces other areas and paradigms of multimedia fingerprinting to show the interrelationship of these works to the goal of the thesis. The review of codebook design is categorized into random, structured, and concatenated classes, depicting a comprehensive history and trend of codebook design. Focus is placed on reviewing code design principles that are effective for fingerprint development. A comparison of these codes is presented in the next chapter, along with the proposed novel code.

Chapter IV, entitled *A Novel Fingerprinting Code*, marks the first novel contribution of this thesis. This chapter presents new code design methodologies, fortifies existing codes that are fundamentally weak, and compares the proposed novel code with selected codes reviewed in the previous chapter. In addition, a new problem, not previously addressed by other work, is presented, for which the novel code is shown to excel.

The novel Joint Source Fingerprinting paradigm is discussed in Chapter V, motivating its need and power. A general analytical description of the JSF methodology is presented. Then a suboptimal JSF algorithm targeted for digital video is presented. In addition, support for a greater number of users, as well as communication broadcasting is discussed for this specific JSF algorithm. Finally simulation results are presented and compared to verify the theoretical observations.

Chapter VI summarizes the achievements in this thesis, and presents future work to extend this research.

CHAPTER II

PROBLEM FORMULATION AND PRELIMINARIES

Traditionally, the problem of digital fingerprinting has been separated into multiple components that are designed and optimized individually [3, 4, 5, 8, 9, 10]. Figure 1 shows a breakdown of the overall problem into sub-problems, presented as the leaves of the tree diagram. The problem of designing a fingerprinting scheme consists of designing a fingerprinting code, and a watermarking scheme that is employed to embed the code into the multimedia data. Use of the fingerprinting scheme in a broadcast channel environment[1] can be conducted in many ways [3, 11, 12, 13, 14, 15] and often depends on the nature of the fingerprinting process. It is therefore natural to consider solving the individual sub-problems in the following order: fingerprinting code, watermarking scheme, and broadcast channel environment.

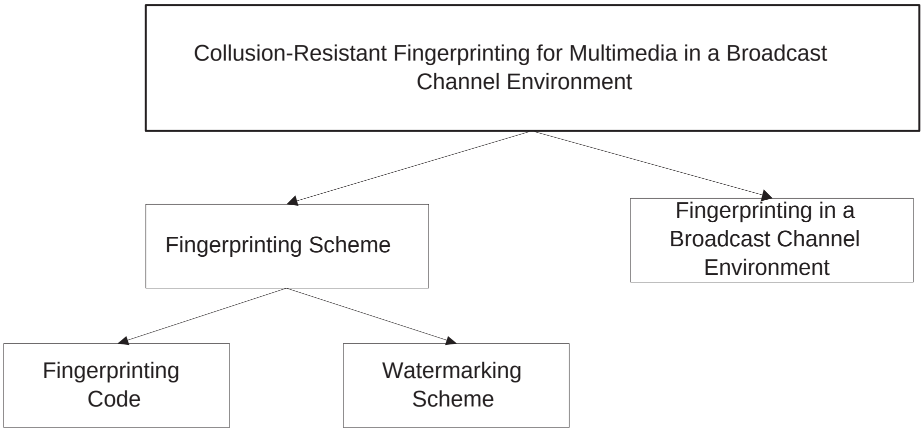

Collusion-Resistant Fingerprinting for Multimedia in a Broadcast Channel Environment
Fingerprinting Scheme
Fingerprinting in a Broadcast Channel Environment
Fingerprinting Code
Watermarking Scheme

Fig. 1. Components of the Problem

Next, each component is mathematically defined.

[1] Recall from Chapter I that broadcasting can help reduce bandwidth usage for applications such as Video on Demand (VoD).

A. Fingerprinting Scheme

A *fingerprinting scheme* consists of two components: the fingerprinting code, and the watermarking scheme.

1. Fingerprinting Code

This section begins with the problem formulation of the fingerprinting code. A *fingerprinting code* consists of a codebook, and a tracing algorithm.

Definition 1 *A codebook is a set* $\Gamma = \{\gamma^1, \gamma^2, \ldots, \gamma^M\} \subseteq \Sigma^l \triangleq \{s_1 s_2 \cdots s_l | s_i \in \Sigma\}$*, of* M *codewords of length* l*, over some finite alphabet* Σ*. Any subset* $\Gamma' \subset \Gamma$ *is also a valid codebook. Also,* γ^i *can be written as* $\gamma^i = \gamma^i_1 \gamma^i_2 \cdots \gamma^i_l$*.*

Definition 2 *A tracing algorithm* $\mathcal{A}$*, is a function* $\Sigma^l \mapsto \mathcal{P}(\Sigma^l)\backslash\{\emptyset\}$*, where* $\mathcal{P}(\Sigma^l)$ *is the power set of* Σ^l*.*

In a fingerprinting scheme, each codeword from Γ is assigned to a different user. The goal of a malicious coalition of users is to combine their codewords to produce a new word η, such that η cannot be traced back to the coalition.[2]

Definition 3 *A successful collusion attack by a coalition of* M *users with codewords in* Γ*, is a function* $\mathcal{Z}$*, such that* $\eta = \mathcal{Z}(\Gamma)$*, and* $\mathcal{A}(\eta) \notin \mathcal{P}(\Gamma)\backslash\{\emptyset\}$*.*

A typical family of candidate attack functions includes the class of Marking Assumption attacks [3] to be discussed later in this chapter. While the adversaries of fingerprinting must design a successful attack function $\mathcal{Z}$, the advocates of fingerprinting must design a codebook, and a tracing algorithm to counter collusion attacks.

[2] The new word η is not necessarily a codeword, because it is possible to have $\eta \notin \Gamma$. Therefore, when referring to a word formed via the collusion attack, the prefix *code* is discarded, and η is just called a *word*.

Definition 4 *The triple* $(\Gamma, \mathcal{A}, \Theta)$*, consisting of codebook* Γ *of cardinality* M*, tracing algorithm* $\mathcal{A}$*, and a family of attack functions* Θ*, is said to be c-collusion-resistant to attacks from* Θ *with* ε*-error* $(0 < \varepsilon \ll 1)$ *if the following properties are satisfied:* $\forall\, \Gamma' \subseteq \Gamma$, $|\Gamma'| \leq c < M$*, let* $\eta = \mathcal{Z}(\Gamma')$ *for* $\mathcal{Z} \in \Theta$*, then* $Pr\,[\mathcal{A}(\eta) \in \mathcal{P}(\Gamma')\backslash\{\emptyset\}] > 1-\varepsilon$*.*

Note, Definition 4 also resists the framing of innocent users, as the tracing algorithm $\mathcal{A}$ will exclusively produce a subset of codewords from Γ', the codebook belonging to the colluders.

2. Watermarking Scheme

A digital watermark is an imperceptible mark, such as a logo, that is embedded into digital media[3]. To quantify the meaning of imperceptibility, a metric $d(\cdot,\cdot) \geq 0$ is defined to measure the similarity between any two multimedia signals C, $\tilde{C}_i$. Let $T > 0$ be a threshold variable such that $d(C, \tilde{C}_1) \gg T$ implies C and $\tilde{C}_1$ are not visually similar, while $d(C, \tilde{C}_1) < T$ implies they are visually similar. For example, $d(C, \tilde{C}_1)$ can represent the Euclidean distance, or mean square error (MSE) between C and $\tilde{C}_1$. The threshold T is dependent on the human visual system for images and video and a function of the human auditory system for audio.

A watermarking algorithm takes a host media, a watermark, a key, and embeds the watermark into the host in an imperceptible manner.

Definition 5 *A watermarking algorithm is a function* $\mathcal{W} : \mathbb{C} \times \mathbb{W} \times \mathbb{L} \rightarrow \mathbb{C}$*, where* $\mathbb{C}$ *is a set of multimedia,* $\mathbb{W}$ *is a set of watermarks, and* $\mathbb{L}$ *is a set of keys. In addition, for any* $C \in \mathbb{C}$*,* $W \in \mathbb{W}$*, and* $L \in \mathbb{L}$*,* $\mathcal{W}(C, W, L)$ *is the watermarked multimedia, such that* $d(\mathcal{W}(C, W, L), C) < T$*.*

[3] In contrast to fingerprinting, watermarking focuses on the signal processing aspects of embedding data in multimedia. As such, issues of collusion and code design are beyond its scope.

Watermarks are extracted from watermarked multimedia via an extraction function $\mathcal{X}$.

Definition 6 *A blind watermark extraction function $\mathcal{X}$, is a function $\mathbb{C} \times \mathbb{L} \mapsto \mathbb{W}$, such that $\mathcal{X}(\mathcal{W}(C, W, L), L) = W$.*

The term blind is traditionally used in the watermarking literature [16] to denote extraction of the watermark without direct reference to the original multimedia C.

The goal of an attacker may be to remove or modify the embedded watermark, hence interfering with its security.

Definition 7 *A feasible attack $\mathcal{F}$ on watermarked multimedia, given $\mathcal{W}$, but not L, is a function $\mathbb{C} \mapsto \mathbb{C}$, such that $d(\mathcal{F}(\mathcal{W}(C, W, L)), C) < T$.*

The attack defined in Definition 7, allows the attacker to have knowledge of the watermarking algorithm, but not the key, which is known as *Kerckhoff's Principle* [16]. In addition, the attack is restricted to have no perceptual effect on the watermarked multimedia preserving, hence its commercial value. A blind watermarking scheme is said to be robust to a particular attack if the extraction function is able to extract the watermark after the given attack.

Definition 8 *A blind watermarking scheme that is strictly-robust to a feasible attack $\mathcal{F}$, is a triple $(\mathcal{W}, \mathcal{X}, \mathcal{F})$, such that $\mathcal{X}(\mathcal{F}(\mathcal{W}(C, W, L)), L) = W$. A blind watermarking scheme that is τ-robust to a feasible attack $\mathcal{F}$, is a triple $(\mathcal{W}, \mathcal{X}, \mathcal{F})$, such that $\mathcal{X}(\mathcal{F}(\mathcal{W}(C, W, L)), L) = \tilde{W}$, and $d(\tilde{W}, W) < \tau$.*

An example of a feasible attack is the additive white Gaussian noise with noise power σ^2 [16, 2]; that is,

$$\mathcal{F}(\mathcal{W}(C, W, L), L) = \mathcal{W}(C, W, L) + n \tag{2.1}$$

where n is an appropriately sized noise vector comprised of Gaussian random processes, with a constant power spectral density (PSD) at σ^2. The attack in Equation 2.1 is made feasible by controlling σ^2.

The watermark W in a fingerprinting system represents information about the fingerprint code in a manner that allows it to be imperceptibly and robustly embedded in the multimedia. In practice, a modulation function is required to map the codewords into unique watermarks [8]. Correspondingly, the demodulation function is used to map watermarks back to codewords [8].

Definition 9 *A modulation function $\mathcal{M} : \Gamma \rightarrow \mathbb{W}$ is a bijection, and its inverse function is the demodulation function $\mathcal{M}^{-1} = \mathcal{B} : \mathbb{W} \rightarrow \Gamma$.*

Definition 9 ties together the fingerprinting code and the watermarking scheme, thus completing the problem formulation of the fingerprinting scheme. The modulation and watermarking process, along with its inverse operations are depicted in Figure 2.

B. Fingerprinting in a Broadcast Channel Environment

The second component of the multi-step fingerprinting scheme is the integration of fingerprinting into a broadcast channel environment. Let C represent the multimedia to be distributed to buyers. In this thesis, C is comprised of a set of N frames, $C = \{C^1, C^2, \ldots C^N\}$ that represents a video sequence.

In a broadcast distribution scenario, confidentiality of the multimedia during transmission through the use of encryption is necessary. Let E, the encryption function, represent an operator that maps $\mathbb{C}$, and a set of keys K, to the set $\hat{\mathbb{C}}$ of encrypted $\mathbb{C}$, such that any $\hat{C} \in \hat{\mathbb{C}}$ cannot be understood by eavesdroppers without the appropriate key.

A particular $\hat{C}$ is sent to M receivers who have legally purchased the video C.

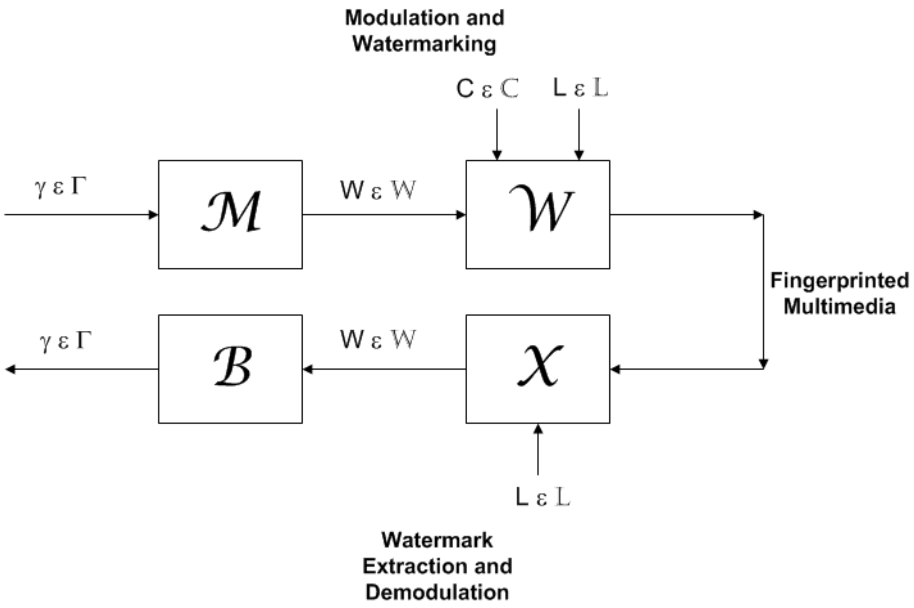

Modulation and Watermarking
C ε ℂ
L ε 𝕃
γ ε Γ
ℳ
W ε 𝕎
𝒲
Fingerprinted Multimedia
γ ε Γ
ℬ
W ε 𝕎
𝒳
L ε 𝕃
Watermark Extraction and Demodulation

Fig. 2. Modulation and Watermarking, and Inverse Operations

Each receiver has unique keys, $K_1, K_2, \dots K_M$. An operator D, the decryption function, maps $\hat{\mathbb{C}}$ and $K = \{K_1, K_2, \dots, K_M\}$, to $\tilde{\mathbb{C}}$, in which $\tilde{C}_i \in \tilde{\mathbb{C}}$ is the fingerprinted version of C, containing the unique fingerprint for receiver i. This scheme is depicted in Figure 3. This architecture is adopted because a single $\hat{C}$ is sent to all users, saving both bandwidth, as well as the complexity that is inherent in schemes that involve sending M unique encrypted content to each user [3, 12].

Referring to the notations introduced in Figure 3, the following requirements are necessary:

(1) **Scrambled video signal:** $d(C, \hat{C}) \gg T$. The encrypted $\hat{C}$ does not visually resemble the unencrypted C, making it unintelligible.

(2) **Unique fingerprinted videos:** $\forall i \neq j,\ d(\tilde{C}_i, \tilde{C}_j) \neq 0$, and $d(C, \tilde{C}_i) < T$. $\tilde{C}_i$ should also contain codewords that are collusion-resistant (Definition 4) and

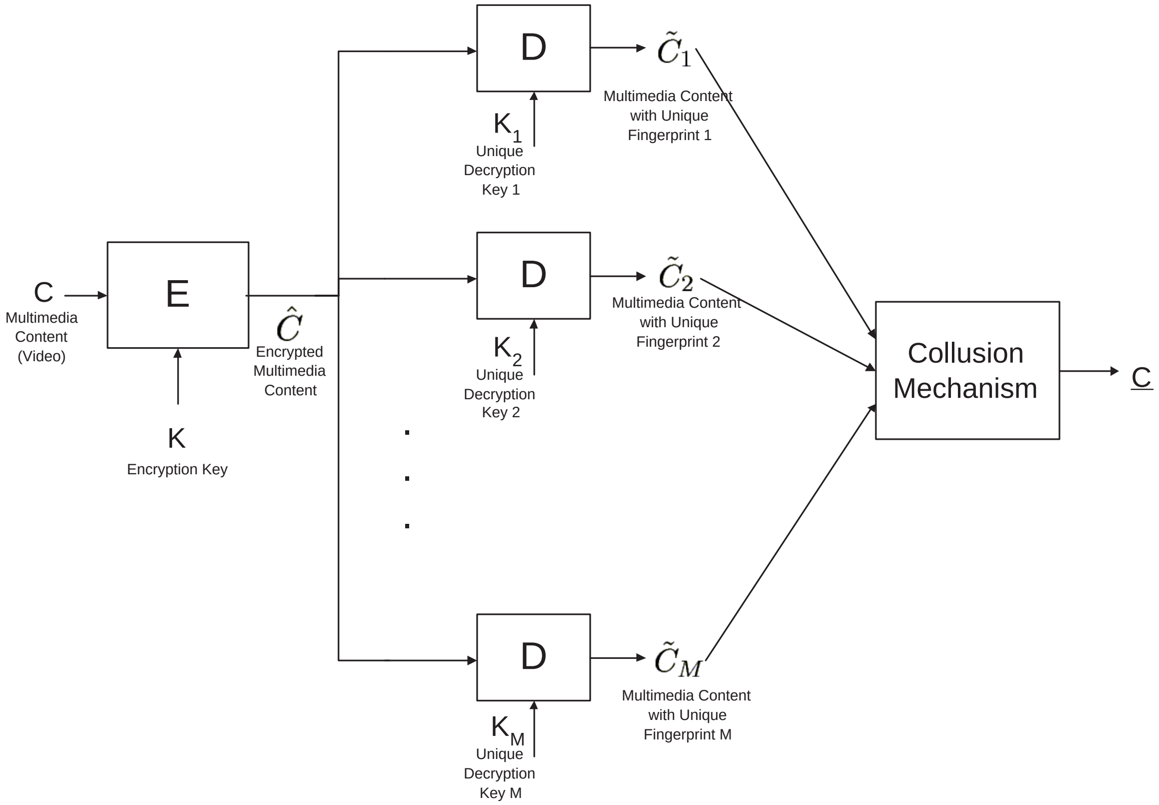

C
Multimedia Content (Video)
E
K
Encryption Key
$\hat{C}$
Encrypted Multimedia Content
D
K1
Unique Decryption Key 1
D
K2
Unique Decryption Key 2
D
KM
Unique Decryption Key M
$\tilde{C}_1$
Multimedia Content with Unique Fingerprint 1
$\tilde{C}_2$
Multimedia Content with Unique Fingerprint 2
$\tilde{C}_M$
Multimedia Content with Unique Fingerprint M
Collusion Mechanism
$\underline{C}$

Fig. 3. Problem Formulation of Fingerprinting in a Broadcast Channel Environment

the watermarking scheme should be robust to a set of feasible attacks, such as AWGN (Equation 2.1).

(3) **Encryption Security:** Without keys $K_1, K_2, \ldots K_M$, an eavesdropper, or the operator D, cannot with computational efficiency [17] derive a $\bar{C}$ given $\hat{C}$, such that $d(C, \bar{C}) < T$.

(4) **Frame-proof:** It is computationally complex to create any set of keys $\{K_i\} \subset \{K_1, K_2, \ldots K_M\}$, given any another set of keys $\{K_j\} \subset \{K_1, K_2, \ldots K_M\}$, where $\{K_i\} \cap \{K_j\} = \emptyset$.

In this thesis, issues of encryption security are not discussed, therefore Criteria 2 is the focus of this thesis.

It should be noted that the proposed scheme is different from the watermarking and multicast schemes in [13, 14] that employ the "trusted" network itself to fingerprint the multimedia in transit. A discussion on the different broadcasting architectures and their pros and cons can be found in [18]. The scheme described in this thesis makes no use of the communications network itself. In addition, the underlying assumption is that each time the transmitter sends data, *all* the users receive exactly the same data; there is no way for the transmitter to only to send data to some users, but not others. This assumption alleviates the restriction of distributing the digital data in question over a communication network, allowing for distribution schemes such as the Compact-Disc (CD) medium; for example, CDs fit into this broadcasting assumption, because they are uncustomized and mass-produced, much like $\hat{C}$ in Figure 3 [3].

A notation that encapsulates the process of modulation, watermarking, collusion, feasible attacks, extraction, and demodulation is now introduced. Suppose $\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_M$ (in Figure 3) are fingerprinted with codewords from Γ. Then define $x = \mathcal{B}(\mathcal{X}(\mathcal{F}(\underline{C}), L))$, where $\underline{C}$ is the multimedia after collusion, as in Figure 3. Γ is called the underlying codebook of x, and the relationship is denoted as $\Gamma \Rightarrow x$. If Γ was not used, then the notation is $\Gamma \nRightarrow x$. This notation is useful as it succinctly ties together the processes from the transmitter, through the attack channel, to the receiver.

C. General Overview of the Complete Problem

The sections above formulated the problem piece by piece. The complete problem is stated in this section. In the multimedia fingerprinting for a broadcast channel environment problem, the goal is to design $(E, D, \{K_i\}_{i=1}^M)$, such that E is the encryption mechanism, D is the decryption mechanism that also results in unique fingerprinted data given unique keys $\{K_i\}_{i=1}^M$. In addition, $(E, D, \{K_i\}_{i=1}^M)$ should adhere to the requirements in Section B. To achieve unique fingerprinted data, $(E, D, \{K_i\}_{i=1}^M)$ has a sub-layer consisting of $(\Gamma, \mathcal{M}, \mathcal{W})$. A receiver is also to be designed, consisting of $(\mathcal{A}, \mathcal{B}, \mathcal{X})$. Ideally, $(\Gamma, \mathcal{A})$ is M-collusion-resistant to attacks generated by higher-level attacks on the multimedia, such as scrambling, and linear estimation and $(\mathcal{W}, \mathcal{X}, \mathcal{F})$ is τ-robust to common single-user attacks, which will be discussed in the next section.

D. Attacks

This section presents 3 types of attacks: codebook attacks, single-user attacks, and multimedia collusion attacks. In codebook attacks, the collusion attacks are modeled as being applied directly to the codewords that are available to a coalition; the media in which the codewords are embedded into are not considered. This is a good starting point in designing a fingerprinting scheme under the multi-step paradigm, as depicted in Figure 1, because it decouples code and watermark design. In single-user attacks, the attack, involving one user and traditionally found in the watermarking literature, is applied directly to the media. Finally, the multimedia collusion attack is applied directly to several copies of fingerprinted multimedia.

1. Codebook Attacks

In [19], to assess the strength of a collusion attack, the word $y = \mathcal{Z}(\Gamma')$, $\Gamma' \subseteq \Gamma$, generated by collusion is generalized and represented by a conditional probability distribution, conditioned on a particular coalition. Here, the generalization will be presented more formally. Let Y be the set of possible attacked words, $G = \mathcal{P}(\Gamma)$, the power set of the codebook Γ, and $\Omega = Y \cup G$. Let $(\Omega, \mathbf{a}, P)$ be a probability space, where $\mathbf{a}$ is a σ-algebra generated from Ω. For $y \in Y$ and $g \in G$, $P(\{y, g\})$ gives the "joint probability" that collusion amongst a set of codewords g will result in a word y. This is the most general representation of an attack, because any word can be generated from any coalition with some non-zero probability. This is however not useful from the point of view of fingerprinting code design, since $P(\{y, g\})$ is not known *a priori*[4].

This unrestricted, general attack presented above is difficult to counter, and hence more restrictive assumptions need to be made concerning what a coalition can and cannot do, given a set of codewords. The most widely used assumption is called the Marking Assumption [3], which is assumed in the majority of the fingerprinting literature, because it allows the fingerprinting problem to be solved from a coding perspective.

Definition 10 (Marking Assumption) *For codebook Γ of cardinality M, the word $\eta = \eta_1\eta_2\cdots\eta_l = \mathcal{Z}(\Gamma)$ conforms to the Marking Assumption when the following condition is true:*

$$\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \Longrightarrow \eta_i = \gamma_i \tag{2.2}$$

Such codeword positions are called undetectable.

Definition 10 states that the positions in the codewords that yield the same alphabet cannot be changed by the coalition.

In Definition 4, Θ is the attack family. For example, any attacks that conform to the Marking Assumption can constitute a family of attacks. Another way to

[4] In [20], $P(\{g\})$, the probability that a coalition with codwords from g will collude, is assumed to be known, and the codes are designed with this knowledge; a tree structure is employed to group together users who are more likely to collude.

encapsulate a family of attacks, is by defining the space in which attacked words can fall in; in [19], this is called the *envelope*.

Definition 11 (Attack Envelope) *The attack envelope of $\mathcal{Z} \in \Theta$, is the range of $\mathcal{Z}$.*

The envelope gives the set of all attacked words, whereas Θ only gives the attack mechanism itself. The definitions that follow, present envelopes used in this thesis that satisfy the Marking Assumption.

A narrow-sense attack belongs to the class of Marking Assumption attacks, and results in words in the narrow-sense envelope (Definition 12), which only allows detectable alphabets to be mapped to alphabets found in the same position.

Definition 12 (Narrow-sense Envelope) *The narrow-sense envelope of $\mathcal{Z}(\Gamma)$ is the set*

$$e(\Gamma) = \{x \mid (\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \implies x_i = \gamma_i) \wedge x_i \in \{\gamma_i^j\}_{j=1}^M\} \tag{2.3}$$

For example, given the words $\Gamma = \{abd, acd, ged\}$, $aad \notin e(\Gamma)$, because the alphabets in the second position are b, c, e, but not a, even though a is detectable in the first position.

A wide-sense attack (envelope given in Definition 13) is similar to the narrow-sense attack, except any detectable alphabet can be mapped to any other detectable alphabet, not necessarily in the same position.

Definition 13 (Wide-sense Envelope) *The wide-sense envelope of $\mathcal{Z}(\Gamma)$ is the set*

$$E(\Gamma) = \{x \mid (\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \implies x_i = \gamma_i) \wedge x_i \in Q \subseteq \Sigma\} \tag{2.4}$$

where Q is the set of detectable alphabets in Γ.

In the previous example, $aad \in E(\Gamma)$, and even $agd \in E(\Gamma)$. So the narrow-sense attack is more restrictive than the wide-sense attack.

In the extended narrow-sense attack (envelope given in Definition 14), the rules are the same as the narrow-sense attack, except any detectable word can also be "erased"; that is the erasure symbol is one that does not belong to the alphabet.

Definition 14 (Extended Narrow-sense Envelope) *The narrow-sense envelope of $\mathcal{Z}(\Gamma)$ is the set*

$$e^*(\Gamma) = \{x \mid (\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \Longrightarrow x_i = \gamma_i) \wedge x_i \in \{\gamma_i^j\}_{j=1}^M \cup \{*\}\} \tag{2.5}$$

where $* \notin \Sigma$.

Finally, the extended wide-sense attack (envelope given in Definition 15) is the most general Marking Assumption attack, allowing detectable alphabets to be mapped to any detectable alphabet, as well as being erased.

Definition 15 (Extended Wide-sense Envelope) *The wide-sense envelope of $\mathcal{Z}(\Gamma)$ is the set*

$$E^*(\Gamma) = \{x \mid (\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \Longrightarrow x_i = \gamma_i) \wedge x_i \in Q \subseteq \Sigma \cup \{*\}\} \tag{2.6}$$

where Q is the set of detectable alphabets in Γ, and $* \notin \Sigma$.

It can be shown [3] that in order to be c-collusion resistant with 0-error, Equation 2.7 must be satisfied: $\forall \Gamma_i \subset \Gamma, |\Gamma_i| \leq c$

$$\bigcap \Gamma_i = \emptyset \Longrightarrow \bigcap E^*(\Gamma_i) = \emptyset \tag{2.7}$$

Unfortunately as shown in [3], this is not achievable with $\epsilon = 0$, when $c \geq 2$, and hence there must be a non-zero error.

2. Single-User Attacks

Feasible attacks on multimedia as defined in Definition 7 are now presented. These attacks involve one copy of the multimedia in question and can be categorized into *unintentional attacks* and *intentional attacks* [2]. Unintentional attacks are those that occur due to bandwidth constraints, such as lossy copying and transcoding (i.e. compression, change in frame rate, format conversion, conversion in display format). Intentional attacks are those user-generated attacks that aim to remove the watermark or fingerprint in the multimedia. Intentional attacks on video can be categorized into *single-frame attacks* and *statistical attacks* [2]. Single-frame attacks can be categorized into *signal processing attacks* (i.e. band-pass filtering, adaptive Wiener denoising [2], etc.) and *desynchronizing attacks* (i.e. affine transformations, scaling, cropping, etc.) [2]. Statistical attacks for video are sometimes also called collusion. However, there is only one copy of the video in question and the term arises from the fact that consecutive frames in the video are used together to remove the watermark or fingerprint. A simple statistical attack on video is to average a small set of consecutive frames in hopes that this will remove the watermark. A more sophisticated statistical attack is to first estimate the watermark in each individual frame and then average the estimated watermarks to obtain a final estimation, which is then subtracted from each frame. Figure 4 shows the classification of single-user attacks. In this thesis, the following single-user attacks are considered: AWGN from Equation 2.1, JPEG compression, small rotations, and translation of random blocks, which will be described in detail later in this thesis.

The next section presents collusion attacks directly on the multimedia, by comparing several copies of the same video with different fingerprints embedded. The attacks involve combining matching frames from multiple copies as opposed to con-

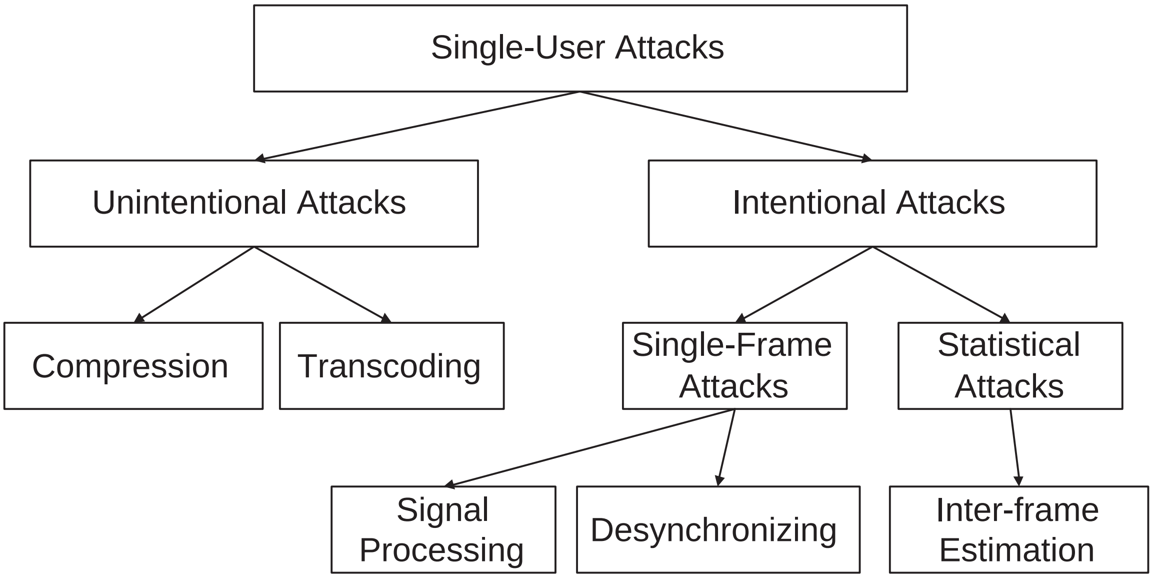


Fig. 4. Single-User Attacks

secutive frames from one copy.

3. Multimedia Collusion

Collusion attacks applied directly on multiple copies of multimedia are now presented. Figure 5 shows one possible classification of collusion on fingerprinted multimedia. When fingerprints are embedded into multimedia, pirates can collude to estimate the original non-fingerprinted multimedia. The collusion attack becomes an estimation problem, and derives many techniques from Estimation Theory. On the other hand, pirates might try to scramble the fingerprint in hopes of framing an innocent buyer, or simply creating a non-compliant fingerprint. The second technique is more *ad-hoc*, and weaker than the first, although it is also computationally cheaper.

Before presenting the attacks, the notation $\underline{C}^j(x,y)$ describes the j^{th} frame at the $(x,y)^{\text{th}}$ pixel of the attacked video $\underline{C}$. The notation $\tilde{C}_i^j(x,y)$ describes the j^{th}

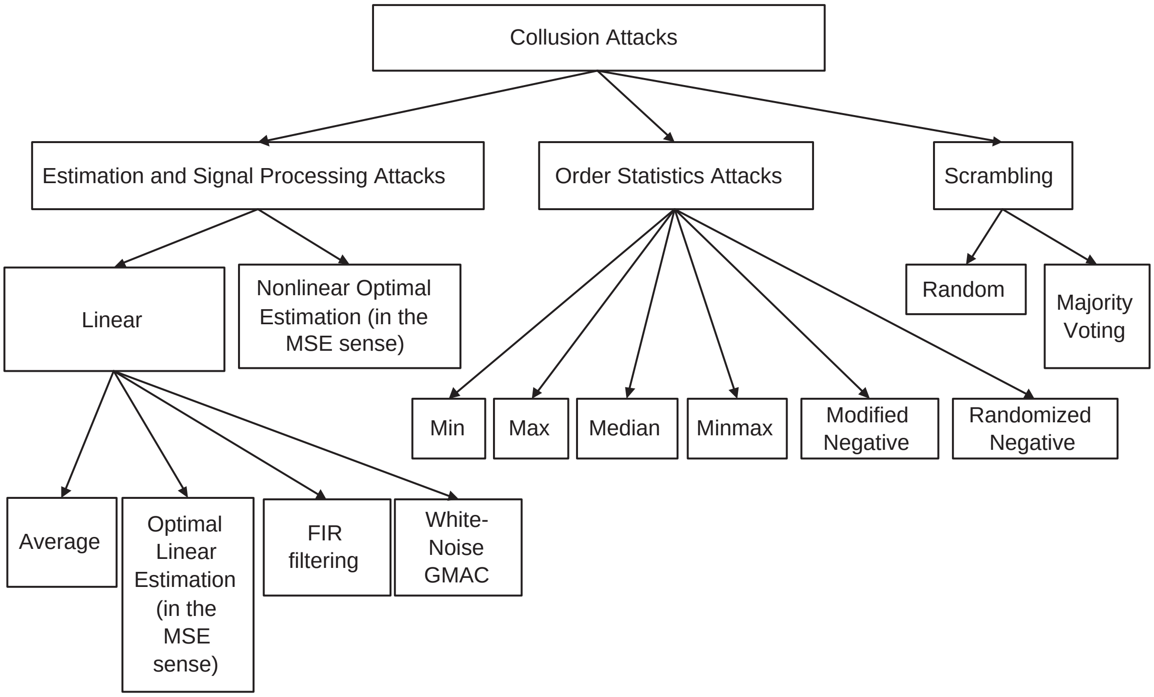


Fig. 5. Types of Collusion on Fingerprinted Multimedia

frame at the $(x, y)^{\text{th}}$ pixel of fingerprinted video for User i.

Equation 2.8 defines the majority attack (which can also be considered an estimation attack, although this relationship is not shown in Figure 5) using the $MODE$ operation from statistics, which chooses the most common value from a set.

$$\underline{C}^j(x,y) = MODE\left(\tilde{C}_1^j(x,y), \tilde{C}_2^j(x,y), \ldots, \tilde{C}_M^j(x,y)\right) \tag{2.8}$$

The random attack selects pixels from the fingerprinted multimedia with equal probability. Equation 2.9 defines the random attack using the $UNIFORM{-}RANDOM$ operation, which selects the indices with equal probability.

$$\underline{C}^j(x,y) = \tilde{C}^j_{UNIFORM-RANDOM(1,2,\ldots,M)}(x,y) \tag{2.9}$$

The simplest suboptimal estimation technique, is simply to average the set of multimedia, as in Equation 2.10.

$$\underline{C}^j(x,y) = \frac{1}{M}\sum_{i=1}^{M}\tilde{C}_i^j(x,y) \tag{2.10}$$

For additional estimation attacks found in Figure 5, the reader is referred to Appendix A.

The order statistics attacks found in [21], consists of the min, max, min max, median, modified negative, and randomized negative defined in Equations 2.11 to 2.16.

$$\begin{aligned}
\underline{C}^j_{\text{min}}(x,y) &= \min\left(\tilde{C}_1^j(x,y), \tilde{C}_2^j(x,y), \ldots, \tilde{C}_M^j(x,y)\right) && (2.11)\\
\underline{C}^j_{\text{max}}(x,y) &= \max\left(\tilde{C}_1^j(x,y), \tilde{C}_2^j(x,y), \ldots, \tilde{C}_M^j(x,y)\right) && (2.12)\\
\underline{C}^j_{\text{med}}(x,y) &= \text{median}\left(\tilde{C}_1^j(x,y), \tilde{C}_2^j(x,y), \ldots, \tilde{C}_M^j(x,y)\right) && (2.13)\\
\underline{C}^j_{\text{min-max}}(x,y) &= \frac{1}{2}\left(\min(\{\tilde{C}_i^j(x,y)\}_{i=1}^{M}) + \max(\{\tilde{C}_i^j(x,y)\}_{i=1}^{M})\right) && (2.14)\\
\underline{C}^j_{\text{mod-neg}}(x,y) &= \underline{C}^j_{\text{min}}(x,y) + \underline{C}^j_{\text{max}}(x,y) - \underline{C}^j_{\text{med}}(x,y) && (2.15)\\
\underline{C}^j_{\text{rand-neg}}(x,y) &= \begin{cases} \underline{C}^j_{\text{min}}(x,y) & \text{with probability } p \\ \underline{C}^j_{\text{max}}(x,y) & \text{with probability } 1-p \end{cases} && (2.16)
\end{aligned}$$

E. The Problem Addressed by This Thesis

In this thesis, the first contribution is in designing $(\Gamma, \mathcal{A}, \Theta)$ for the extended narrow-sense attack, and also providing a means to extend existing codes that are less robust, in such a way that they become robust to the the extended narrow-sense attack. At the same time, the code should have relatively short codeword length[5] for large coalition sizes. In previous works, the codeword length grows very fast when coalition

[5] The reason why minimizing codeword length is important, is because a codeword may not embed into an image of video if it is too long.

sizes are large. The goal of the proposed code is to have the codeword length grow slower than existing codes, when all parameters except the coalition size is constant.

The novel Joint Source Fingerprinting paradigm, constituting the second major contribution of this thesis will develop a framework that integrates all the steps presented above into one design step, and at the same time offer a new feature to deter the collusion process. A suboptimal algorithm for video fingerprinting is to be designed, such that the algorithm is robust to collusion from small coalition sizes. The design should be resilient to the following collusion attacks:

(1) Average attack from Equation 2.10;

(2) Random attack from Equation 2.9;

(3) Order Statistic attacks from Equations 2.11 to 2.16.

The algorithm should be resilient to the following single-user attacks:

(1) AWGN from Equation 2.1;

(2) JPEG compression;

(3) Small rotations;

(4) Translation of random blocks.

When collusion attacks are applied to a video that is fingerprinted using the proposed algorithm, the attacked video should exhibit visual degradation, which is in contrast to existing fingerprinting schemes, specifically those that are watermarked in the manner of [1, 22, 8]. In addition, the proposed algorithm should exhibit qualities that lend itself to efficient broadcasting under the assumptions stated earlier. Specifically, the proposed algorithm should be superior to [8, 11] in terms of bandwidth efficiency.

CHAPTER III

LITERATURE REVIEW

The fingerprinting schemes discussed in this thesis are known as symmetric fingerprinting. In symmetric fingerprinting, the buyer has no protection against a dishonest merchant whose intent is to frame the innocent buyer. This problem led to a solution known as asymmetric fingerprinting [23], allowing the buyer to avoid being framed by dishonest merchants. The major drawback in that scheme is the amount of personal information that the buyer needs to disclose to merchants. The solution to this problem is known as anonymous fingerprinting [24], where there is a third party devoted to collecting and keeping personal information private. These two fingerprinting schemes are protocol-heavy, and some existing symmetric fingerprinting schemes can be modified to adopt these protocols, hence inheriting their frame-proof and privacy advantages. The next section gives a summary of symmetric multimedia fingerprinting techniques to show the interrelationship of these works to the goal of the thesis.

A. Classification of Multimedia Fingerprinting

Symmetric fingerprinting can be classified by how the fingerprint is embedded. Figure 6 shows some existing fingerprinting techniques.

The majority of symmetric fingerprinting schemes embed the fingerprint imperceptibly into the content, and in Figure 6, this is categorized as *Fingerprinting Using Codebooks and Watermarking*, which is also presented in the Problem Formulation chapter. Methods for embedding a fingerprint codeword into multimedia can be found in the watermarking literature [1, 22, 8, 25, 16, 16, 26]. Within the realm of Fingerprinting using Codebooks and Watermarking, some works focus on the fingerprinting

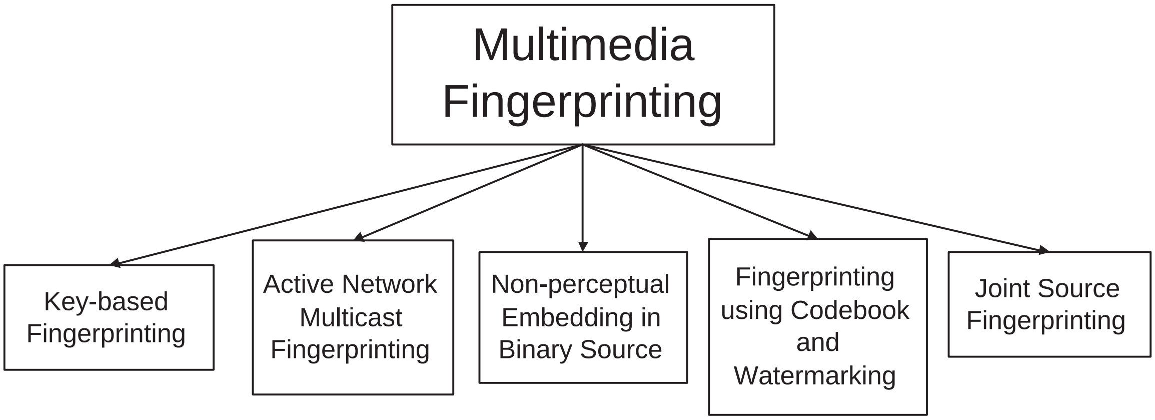

Multimedia Fingerprinting
Key-based Fingerprinting
Active Network Multicast Fingerprinting
Non-perceptual Embedding in Binary Source
Fingerprinting using Codebook and Watermarking
Joint Source Fingerprinting

Fig. 6. Fingerprinting Techniques

codebook itself, with little reference to embedding, while others focus more on embedding, as well as advanced watermark detection, as in the works of [26]. Within the codebook research community, different emphasis is placed on criteria such as attack assumptions, coalition size, growth in probability of error, growth in codeword length, ability to trace one or all pirates, etc. Most researchers only concentrate on one of these criteria. For example, historically, many papers have been written on fingerprinting codes that are only secure against a coalition size of 2 or 3 pirates, such as [9, 6, 27]. Even in a more recent paper [19], much emphasis is given to collusion-resistance against 2 pirates. In this thesis, emphasis will be placed on coalitions of much larger sizes, such as some non-negligible percentage of the total number of users.

Some fingerprinting schemes, such as [28, 29], do not embed the fingerprint in the actual content, and the fingerprint is really a unique key for applications such as pay-TV decryption. In Figure 6, this is categorized as *Key-based Fingerprinting.* Most Key-based Fingerprinting schemes aim to create codes that are resilient to at most narrow-sense attacks; that is, erasure of detectable marks is not allowed, and also mapping any detectable bit to alphabets not in that particular bit position is prohibited. Since the code is really a key, this makes sense, as erased bits would prove useless in a key. These codes are called *traceability codes* (the term *Tracing Traitors* is often found in the title of such works), and can be found in [28, 30, 31]. In [10] traceability codes are extended to allow a set number of erasures, by using the Guruswami-Sudan soft-decision decoder. However the number of allowed erasures is usually not enough to make the code resilient to extended narrow-sense attacks, as will be shown by this thesis. In addition, this thesis will offer an alternative way to extend traceability codes in such a way that the resulting code is resilient to the extended narrow-sense attack.

In [29], the system embeds one copyright watermark in all copies, and a unique key is embedded in the users' media player. The assumption is that the watermarked video can only be played in the proprietary player, which checks the unique user key with the embedded watermark to see if the user has rights to the content. If a malicious user is able to obtain his unique key from the media player, he can remove part of the watermark corresponding to his player, hence fooling his media player into thinking that there is no copyright watermark in the video. However, since only part of the watermark is removed, the missing part provides information that can be used to trace the malicious user, i.e. the partial watermark becomes a fingerprint. To break this system, the video is either converted to another format and played with another player that does not check for this watermark, or the malicious user can record the video from the information being sent to his video card, thus avoiding the media player altogether.

As noted earlier, the communication network can actively participate in fingerprinting, by selectively dropping packets, such that the resulting sequence of packets at one user's receiver differs from the sequence of packets at other users' receivers. This uniqueness can be used to trace a malicious user who illegally re-distributes his media. Such schemes can be found in [13, 14], and are referred to as *Active Network Multicast Fingerprinting* in Figure 6.

Early works on fingerprinting did not make use of the media itself, but instead embedded fingerprints in the binary source, i.e. the bit representation of the media. The main drawback with this method is that random perturbation of the binary source can easily result in visual or audio degradation. In addition, perturbation of bits that are safe against degradation, such as the least significant bits, can easily be removed by an adversary, since these bit locations are known to be safe against any perturbations, including malicious ones. The lack of integration of the fingerprinting code with the media content itself, made this method of fingerprinting, termed *Non-perceptual Embedding in Binary Source* in Figure 6, obsolete very quickly. Works such as [19, 12] adhere to this method of fingerprinting. Although this method of embedding is weak, insight on code design can be drawn from these works, and hence they are reviewed in this paper.

Finally the *Joint Source Fingerprinting* paradigm will be introduced in this thesis. This concept is fundamentally different from existing fingerprinting techniques, in that there is a higher integration of the fingerprinting codes with the media. In fact, the source media itself is used as the codebook alphabet, instead of creating a codebook from independent alphabets not related to the source.

The proceeding sections will delve into specific instances of codebook design in symmetric fingerprinting. The emphasis is in showing a trend in fingerprinting codebook research; that is, the beginning of fingerprinting research started with codes generated randomly, progressed to codes that have deterministic structure, and finally led to the present day concatenated codes that incorporate structure and randomness, along with error-correcting capabilities.

B. Limitations of Digital Fingerprinting

It is no surprise that with a sufficient number of colluders, any fingerprinting scheme can be defeated. For example, in [32], it was shown that with $\Omega(\sqrt{n \ln m})$ fingerprinted copies, where there are m copies containing unique i.i.d. Gaussian watermarks of length n, colluders can successfully erase the fingerprints. In many cases, there is an upper bound on the number of colluders that can be handled by a collusion-resistant code, before the fingerprints can be erased. However, these findings do not suggest that the fingerprinting research exists in vein. For example, in [4, 20], group-based construction of collusion-resistant codes is used to group buyers suspected of being more likely to collude with one another. The goal of fingerprinting research is to deter, but not to prevent pirates who spend more time and money to erase the fingerprints.

C. Random Codes

1. Chameleon Cipher

In an attempt to merge fingerprinting and decryption to satisfy the broadcasting requirements of Section B also depicted in Figure 3, the Chameleon Cipher was introduced [12]. In this section, the coding method is studied, and ideas from this work are later used to propose a novel fingerprinting code in Chapter IV.

The Chameleon cipher is collusion-resistant to a coalition size of at most 4 pirates. The code is generated randomly, and the idea is that any two users are expected to have at least one bit in common, and these common bits will be unique to each pair of users. Hence, when a coalition of 2 pirates colludes, and abides by the Marking Assumption, these common bits will uniquely identify the pirates. When 3 pirates collude, and employ the majority attack (Equation 2.8), common bits between each pair of users will again leave a traceable trail. Even when 4 pirates collude, and employ a random attack (Equation 2.9), these common bits will remain with high probability. Therefore, it is the common bits between pairs of users that identify pirates in this scheme.

The next section presents codes that have structure, such that the shortcomings of randomness are alleviated.

D. Structured Codes

The concept of having common bits between some codewords, but not others, as introduced by the Chameleon Cipher in the previous section, is exploited in structured codes. The difference between the structured codes presented in this section, and the Chameleon Cipher, is that structured codes explicitly introduce common bits between different codewords, as opposed to relying on randomness to achieve this.

1. Fingerprinting Long Forgiving Messages

In [33], unique common bits between unique coalitions define the recipe for creating fingerprinting codes. Let N be the number of bits in the media, in which the fingerprints are to be embedded. For all subsets $A \subset \{1, 2, \ldots, M\}$, $|A| \leq k$, choose a subset $S(A) \subset \{1, 2, \ldots, N\}$, such that Equation 3.1 is satisfied.

$$A \neq B \Longrightarrow S(A) \cap S(B) = \emptyset \tag{3.1}$$

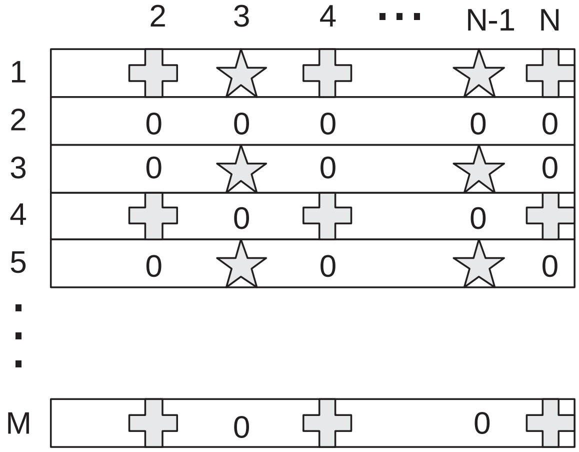
Then the fingerprints F_i are formed according to Equation 3.2.

$$F_i(j) = \sum_{i \in A} \chi_{S(A)}(j) \tag{3.2}$$

for $j \in \{1, 2, \ldots, N\}$, and $\chi_{S(A)}(j)$ is the characteristic function:

$$\chi_{S(A)}(j) = \begin{cases} 1 & \text{if } j \in S(A) \\ 0 & \text{if } j \notin S(A) \end{cases} \tag{3.3}$$

The fingerprints are then added bit-wise to the media C, resulting in $\tilde{C}_i = C + F_i$, where $\tilde{C}_i$, C, F_i are all in bit representation form.



	2	3	4	⋯	N-1	N
1	✚	☆	✚		☆	✚
2	0	0	0		0	0
3	0	☆	0		☆	0
4	✚	0	✚		0	✚
5	0	☆	0		☆	0
⋮						
M	✚	0	✚		0	✚

Fig. 7. Example of Fingerprinting Code Construction

Figure 7 depicts an example of the fingerprint code construction. Suppose there are M users with codewords represented by each of the M rows. Let $A = \{1, 4, M\}$ and $B = \{1, 3, 5\}$, so $A \neq B$ (i.e. two disjoint coalitions). Let $S(A) = \{2, 4, N\}$ (crosses) and $S(B) = \{3, N-1\}$ (stars), so $S(A) \cap S(B) = \emptyset$ (i.e. the bit positions are disjoint). The crosses and stars represent 1's in the fingerprint codeword. Note that the bits in the same bit position (i.e. same column) that are not crosses nor stars, are explicitly set to 0's.

If a majority attack (Equation 2.8) occurs between users $1, 3, 5$, then bits 3, and $N-1$ will remain as 1. The fact that both bits $3, N-1$ are 1's can be used to identify the culprits. However, bits $2, 4, l$ can also be set to be 1's. In this case, user 1 can be identified with confidence as a culprit, but there would be confusion as to whether users $1, 4, M$ are culprits or not. The shortcomings of this fingerprinting construction are:

(1) The fingerprint is embedded via XOR with the binary representation of the multimedia, and hence this may cause visual distortion if certain bits in the multimedia are altered.

(2) As noted in [33], the fingerprint codeword length grows exponentially in the maximum size of the coalition, and hence the coalition size must be relatively small.

2. Projective Geometric Codes

In [8], a fingerprint is a collection of marking positions that are either marked with 1, or not marked, being equivalent to 0. The idea is to construct fingerprint codewords that "intersect" with other fingerprint codewords in unique ways. Assuming that the unique intersections between unique coalitions of pirates cannot be changed, as in the Marking Assumption, these unique intersections will determine all the colluders.

The concept of unique intersections has a nice geometric interpretation. For fingerprinting codes that can detect at most 2 colluders, the codewords that make up Γ can be represented by the edges on the triangle in Figure 8. Any two users have a unique intersection at the vertices of the triangle. If the pair of users remove their detectable marks on the edge of the triangle, the intersection will remain intact, revealing the identities of the 2 colluders. If the colluders do not remove all the detectable marks, i.e. there is some leftover edge, then the leftover edge can be used to trace the colluders as well. Hence it is in the best interest of colluders to remove

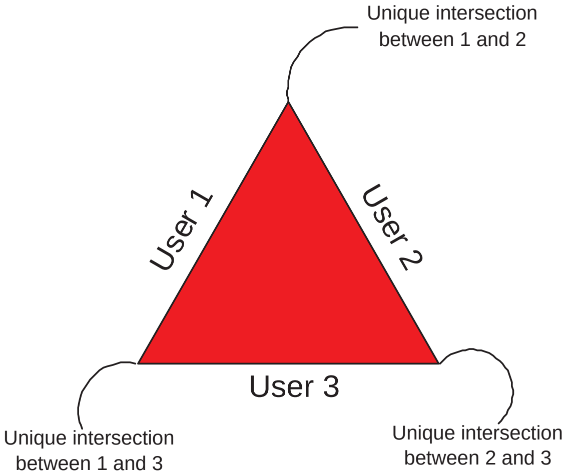

Unique intersection between 1 and 2
User 1
User 2
User 3
Unique intersection between 1 and 3
Unique intersection between 2 and 3

Fig. 8. Geometric Interpretation of 2-collusion-resistant Codewords

detectable marks.

A possible attack that can cripple this system, is to remove the edges, but leave all vertices intact. However, if the codeword is modulated and watermarked into the media using a secret key, the attackers do not know where the verticies and edges are, hence they can only guess. As will be seen later, when the geometric shapes reside in higher dimensions, it will be more difficult for the colluders to identify the vertices and edges.

Figure 9 depicts a tetrahedron, where the 4 sides represent the codewords for 4 users that can trace at most 3 colluders. When 2 users collude, they share a unique edge. When 3 users collude, they share a unique vertex.

For codewords that can detect at most 4 colluders, a geometric shape in 4-dimensions is used. In general, codewords that can detect at most n colluders, require shapes in n-dimensions. The hyperplanes of the higher-dimension "hyper-tetrahedron", represents the codewords. These geometric shapes are coded using the theory of projective geometry, and the details can be found in [34, 35]. The main

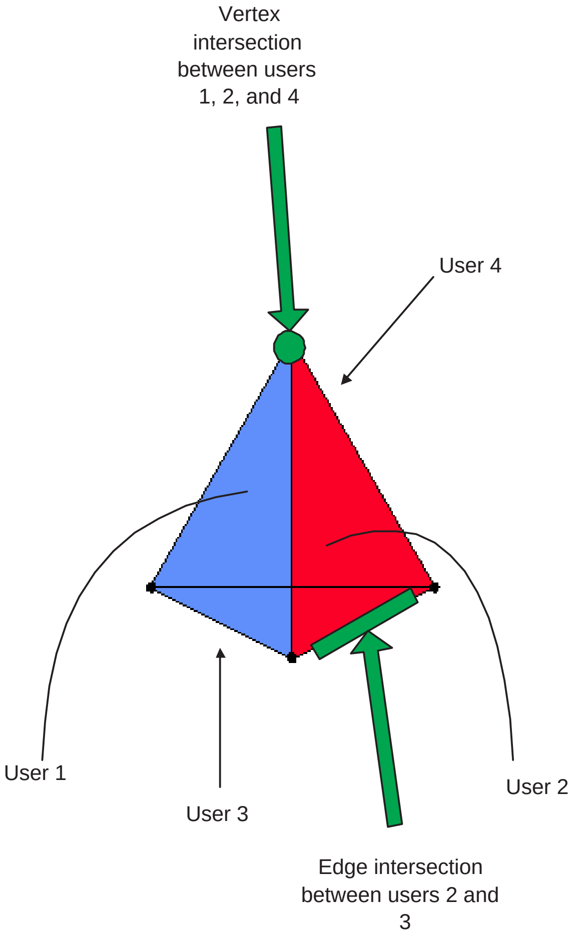

Vertex intersection between users 1, 2, and 4
User 4
User 1
User 3
User 2
Edge intersection between users 2 and 3

Fig. 9. Geometric Interpretation of 3-collusion-resistant Codewords

shortcoming of this approach is that the length of the codewords is $O(M^c)$ bits, where M is the total number of codewords (i.e. total number of users), and c is the maximum coalition size that can be supported.

3. Balanced Incomplete Block Design Codes

The idea of using the unique intersection of codes to determine colluders is also used in [36, 4, 26]. Instead of taking a geometric approach, Γ is designed using balanced incomplete block design (BIBD) from combinatorial design theory [37, 38, 39]. One of the drawbacks of the BIBD codes is that they are only resilient to the binary AND operator. That is, when a coalition applies the binary AND operation to their codewords, the resulting word will differ from a word created from a different set of words. Therefore, different attacked words will "intersect" at different bit positions when the attack is the AND operation. However, under other collusion attacks, this scheme cannot be used to trace pirates.

Another drawback of using the BIBD code, is that they do not exist for any arbitrary number of users M, and coalition size c. If a symmetric (l, c, λ) BIBD code exists, $M = \lambda \frac{l^2-l}{c^2-c}$ users are supported, with coalition size at most $c-1$. Furthermore, the length of each codeword is l, so the length is approximately $O(c\sqrt{M})$ bits. The conditions on the existence of BIBD-codes can be found in [37, 38, 39].

Although the codeword length is much shorter than those of the codes previously reviewed, the shortcoming of the BIBD codes is that they are only robust to the AND operator.

E. Concatenated Codes

The latest trend in fingerprinting codes is the use of *concatenated codes.* The idea behind concatenated codes is that codes that are weak[1] can be reinforced by using an error correcting code. The method by which fingerprinting codes and error correcting codes are combined is called *concatenation.* Concatenated codes are defined in Definition 16.

Definition 16 (Concatenated Codes) *Let V be a codebook of size q. Let W be a*

[1] This weakness can be the lack of robustness to collusion attacks, or the lack of robustness to larger coalitions [3], as well as an unacceptable probability of error [19].

q-ary code. Then a (V, W) concatenated code is the following

$$\{\varphi(w_1)\varphi(w_2)\cdots\varphi(w_l) \mid w_i \in W,\ \varphi : W \to V\} \tag{3.4}$$

and φ is a one-to-one correspondence.

Two works use concatenated codes to achieve shorter fingerprinting codeword lengths, and at the same time maintain a small probability of error.

1. Fingerprinting Under the Marking Assumption

The construction of fingerprinting codes in [3] is fundamentally different from those codes previously reviewed, in that unique intersections are not used to identify pirates, but rather, the distribution of bits in the attacked word is used.

The codebook design is now described. Let $|\Gamma| = M$, and construct the $M \times (M-1)$ matrix:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Each column is replicated $d = 2M^2 \log(2M/\epsilon)$ times, where ϵ denotes a user-defined probability of error. After each column is replicated d times, a random permutation of the columns of the matrix is then performed. The M rows of this resulting matrix make up the codewords in Γ, and are assigned to M different users. The code is able to resist collusion up to M users.

The tracing algorithm is now described. Suppose η is the word created from a collusion attack. Let B_m be the m^{th} block of d bits from the unpermuted η (that is the reverse permutation from the construction of Γ is applied), where the B_m's are non-overlapping. Let $R_s = B_{s-1}B_s$ for $s \in \{2, 3, \ldots, n-1\}$. Let $weight(x)$ be the number of 1's in x, where x is a word made up of bits. The tracing algorithm $\mathcal{A}$, is then:

```
(1) If $weight(B_1) > 0$ then output "User 1 is guilty".
(2) If $weight(B_{n-1}) < d$ then output "User $n$ is guilty".
(3) For all $s = 2$ to $n - 1$ do:
    Let $k = weight(R_s)$.
    If $weight(B_{s-1}) < \frac{k}{2} - \sqrt{\frac{k}{2}\log\frac{2n}{\epsilon}}$ then output "User $s$ is guilty".
```

In addition, the unreadable marks are set to 0.

It can be seen from the tracing algorithm that the distribution of 1's in the attacked word is used to determine the pirates. This is fundamentally different from the tracing algorithm in the works previous reviewed, because the tracing algorithm in those works used fixed common bits to identify the pirates.

The concept of concatenating fingerprinting codes is used by [3] to shorten the length of their fingerprinting codes, but at the same time robustness to larger coalition sizes is sacrificed. Here, the inner code V is the c-collusion-resistant code previously presented. The outer code W consists of codewords generated randomly with uniform distribution. The tracing algorithm for this concatenated code can be found in [3].

The length of this code is

$$l = O\left(c^4 \log\left(\frac{M}{\epsilon}\right) \log\left(\frac{1}{\epsilon}\right)\right) \tag{3.5}$$

where c is the maximum coalition size, M is the total number of users supported, and ϵ is the probability of error.

2. Separating Codes

In the concatenated fingerprinting code of [19], the inner code V is a (c,c)-separating code, and the outer code W is a $(N,K,\Delta=\delta N)$ linear code.

Definition 17 **((t,t')-separating Code)** *A code V is a (t,t')-separating code when for $X\subset V$, $Y\subset V$, $|X|=t$, $|Y|=t'$ the following holds:*

$$X\cap Y=\emptyset \Longrightarrow e(X)\cap e(Y)=\emptyset \tag{3.6}$$

where e is the narrow-sense envelope.

Equation 3.6 is a weaker condition than the condition of 0-error collusion-resistance given in Equation 2.7. For example, the condition of being pair-wise disjoint is less stringent than being disjoint for more than two sets. Therefore an outer code is needed to fortify separating codes.

The probability of error bound as given by [19], is reproduced in Equation 3.7.

$$\epsilon \leq 2^{-nR(V)\left((\log_2 q)^{-1}D\left(\sigma\|\frac{c-1}{q-1}\right)-R(W)\right)} \tag{3.7}$$

Here V is a (c,c)-separating binary (m,q) code, where $q=|V|$, m is the length of a codeword in V, and $R(V)=\frac{\log_2(q)}{m}$ is the rate of V; W is a $(N,K,\Delta=\delta N)$ linear error code, $R(W)=\frac{K}{N}$ is the rate of W;

$$\delta > 1-\frac{1}{c^2}+\frac{c-1}{c(q-1)},$$

$$\sigma=\frac{1}{c}-(1-\delta)c;$$

$D\left(\sigma\|\frac{c-1}{q-1}\right)$ is the relative entropy defined as

$$D(\sigma\|p)=\sigma\log_2\left(\frac{\sigma}{p}\right)+(1-\sigma)\log_2\left(\frac{1-\sigma}{1-p}\right);$$

finally the entire code is of length $n = mN$, supporting up to q^K users, and being robust to coalitions of up to size c. The length of the code is approximately given by Equation 3.8, which is derived in Appendix B.

$$l = O\left(2^c \log_2(M) \log_2\left(\frac{M}{\epsilon}\right)\right) \tag{3.8}$$

The next section describes a simple broadcasting scheme that operates under the assumptions described earlier.

F. A Simple Broadcasting Scheme

Most of the broadcasting schemes in the literature do not meet the assumptions in Section B. Many of them use more than one channel, or use some multicast scheme. In this section, one broadcasting scheme from [3, 40] that fits the requirements of Section B is presented.

A plaintext object P of length L, is partitioned into l pieces. These l pieces are then replicated so that there are two copies of the l pieces. The first l pieces are embedded with a 0 using some watermarking technique, and the second set of the l pieces is embedded with a 1. Each of the $2l$ pieces is encrypted with $2l$ distinct keys. Each user then receives all $2l$ pieces, but only l keys that can decrypt one of the two pairs. The l keys that are distributed to users correspond to a binary fingerprint codeword, hence if collusion of keys occurs, the resulting output is traceable. Figure 10 depicts the process of partitioning, embedding, and encrypting. This broadcasting scheme requires the transmission of 2 times the size of the original object; that is the size of transmission is $2L$, so the transmission size is $O(1)$ with respect to the number of users M, that the copies will be distributed to. In a traditional many-to-many distribution where every unique user receives a unique copy, the transmission

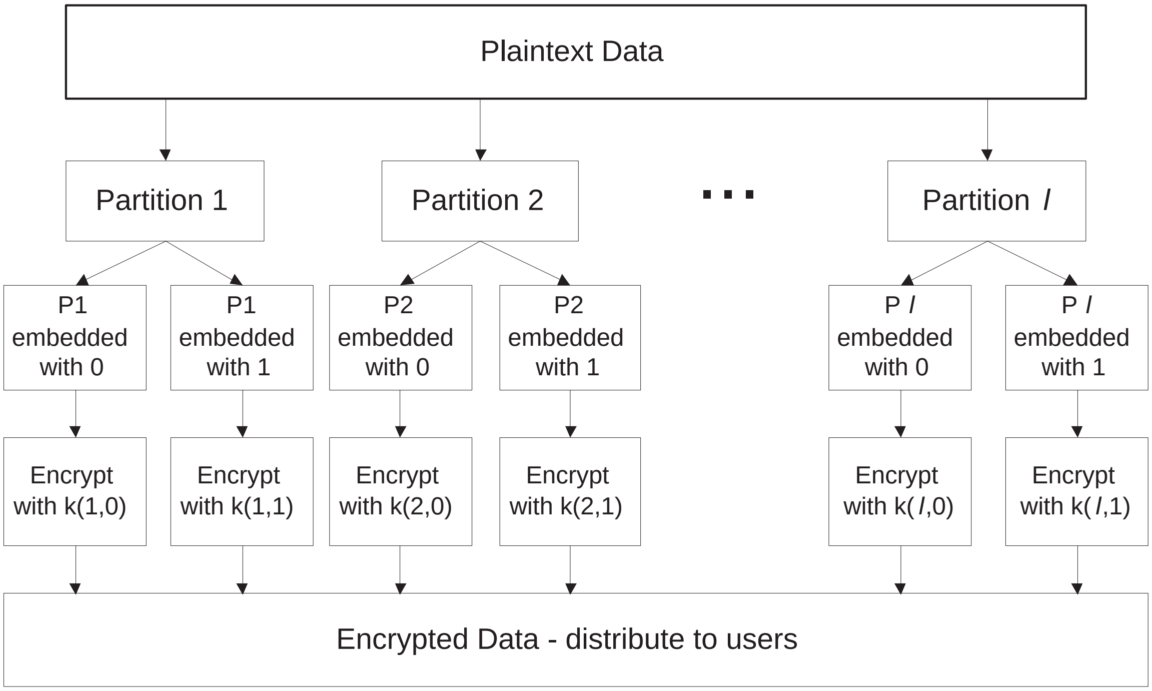


Fig. 10. Distribution of Encrypted Data - A Broadcast Channel Approach

bandwidth grows linearly with the number of users; that is the size of transmission is $M \times L$ for M users, so the transmission size is $O(M)$.

The following chapter introduces a novel fingerprinting code that uses common bits between codewords, as well as the bit distribution in the attacked word, to trace pirates. In addition, concatenation is used, but the roles of the inner and outer codes are reversed. In essence, the ideas from the works reviewed earlier are all integrated to give rise to this new fingerprinting code. It will be shown that this novel scheme can be used to improve the robustness of weak codes, such as the code found in a recent paper [4].

CHAPTER IV

A NOVEL FINGERPRINTING CODE

In the works of [12, 33, 8, 4] common bits between sets of codewords are used to trace pirates. In the concatenated fingerprinting codes of [3, 19], the inner code has some type of weak collusion-resisting capability, while the outer code is an error-control code. In this chapter, a novel fingerprinting code design that uses both common bits and concatenation is presented. The roles in concatenation are reversed, such that the inner code is an error-control code, while the outer code has some type of collusion-resisting capability. The resulting code is then merged with a traceability code to achieve robustness against narrow-sense attacks. The spirit of this work is that weak codes can be combined and result in stronger codes.

A. Step 1: The Outer Code

The code design first starts with a code that is only robust to the hard extended attack, whose envelope is given by Definition 18.

Definition 18 (Hard Extended Envelope) *The Hard Extended Envelope of a codebook* $\Gamma = \{\gamma^1, \gamma^2, \ldots, \gamma^M\} \subset \Sigma^l$*, is the set*

$$E^{h*}(\Gamma, h) = \{x_1 x_2 \cdots x_l \mid (\gamma_i^1 = \gamma_i^2 = \cdots \gamma_i^M \implies x_i = \gamma_i) \wedge (\exists j, k\ \gamma_i^j \neq \gamma_i^k \implies x_i = h)\} \tag{4.1}$$

where $h \in \Sigma \cup \{* \mid * \notin \Sigma\}$ *is fixed.*

Indeed $E^{h*}(\Gamma, h) \subset E^*(\Gamma)$ for $h \in \Sigma \cup \{* | * \notin \Sigma\}$, so Definition 18 is a more restrictive attack than the extended wide-sense attack. Essentially, pirates always replace a detectable alphabet with a fixed symbol h.

The following lemma is implicitly used by the tracing algorithms in [12, 33, 8, 4].

Lemma 1 $\forall\ \Gamma', \Gamma'' \subset \Gamma,\ |\Gamma'| \leq c,\ |\Gamma''| \leq c$

$$\Gamma' \neq \Gamma'' \Longrightarrow \mathcal{Z}(\Gamma') \neq \mathcal{Z}(\Gamma'') \tag{4.2}$$

when $\mathcal{Z}(\Gamma) \in E^{h*}(\Gamma, h)$ *for a fixed* h *is a sufficient condition for c-collusion-resistance as defined in Definition 4.*

Proof 1 *Since every unique coalition of colluders maps to another unique attacked codeword, the function* $\mathcal{Z}$ *is injective. Therefore the image of any* Γ' *can always be mapped back to its unique pre-image* Γ'*. The tracing algorithm thus maps* $\mathcal{Z}(\Gamma')$ *back to* $\mathcal{A}(\mathcal{Z}(\Gamma')) = \Gamma'$*. The probability of error is thus* 0*, and hence*

$$Pr[\mathcal{A}(\mathcal{Z}(\Gamma')) \in \mathcal{P}(\Gamma')\ \{\emptyset\}] = 1$$

satisfying Definition 4.

□

Lemma 1 is analogous to the idea of achieving the information channel capacity for the noisy typewriter [41]. Figure 11a shows a noisy typewriter. There is some probability that given A was sent, the receiver receives either A or B with equal probability. Each alphabet has a possible chance of being confused with its increasing adjacent alphabet. To achieve information channel capacity, that is maximum transmission with low error, only a subset of the inputs are used as shown in Figure Figure 11b. The channel then becomes injective or one-to-one, and the receiver can always decide which alphabet was sent. This is the exact idea adopted in Lemma 1; by designing codebooks with codewords such that their attacked counterpart does not coincide, the tracing algorithm can always decide which codewords were sent. However, Lemma 1 is not achievable with any attack $\mathcal{Z}$, and hence it is only stated for the hard extended attack.

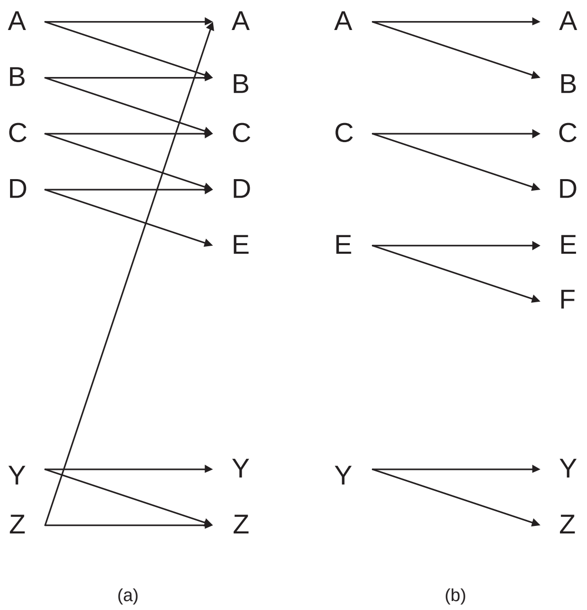


Fig. 11. (a) Noisy Typewriter; (b) Noiseless Typewriter with Only a Subset of Inputs

The codes in [4, 36] do in fact satisfy Lemma 1 for $h = 0$, but not $h = 1$. For example, a code given in [4, 36] is

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

where each row represents a codeword. Under the hard extended envelope $E^{h*}(\Gamma, 1)$, $Z(\{\gamma^1, \gamma^2\}) = Z(\{\gamma^1, \gamma^4\})$, however $\{\gamma^1, \gamma^2\} \neq \{\gamma^1, \gamma^4\}$, therefore Equation 4.2 is not satisfied. The next step is to extend this code such that it is robust to the hard extended attack for $h = 1$.

The following lemma describes a means to improve the robustness of codes that are only robust against one fixed symbol in the hard extended attack.

Lemma 2 *For the binary alphabet $\Sigma = \{0, 1\}$, let $\Gamma = \{\gamma^1, \gamma^2, \ldots, \gamma^M\}$ be a codebook that satisfies Equation 4.2 for one h, that is either $h = 0$, or $h = 1$, but not both. Then*

(1) Γ also satisfies Equation 4.2 for $h = *$.

(2) Let $\Gamma\& = \{\gamma\overline{\gamma} \mid \gamma \in \Gamma\}$, where $\overline{\gamma}$ is the bit complement of γ. Then $\Gamma\&$ satisfies Equation 4.2 for $E^{h*}(\Gamma\&, 0) \cup E^{h*}(\Gamma\&, 1) \cup E^{h*}(\Gamma\&, *)$.

Proof 2 *Suppose without loss of generality that Γ satisfies Equation 4.2 for $h = 0$, but not $h = 1$. The words in $E^{h*}(\Gamma, 0)$ uniquely identify unique coalitions. If any of the 0's in theses words are replaced by $*$, the words are still unique, since the distribution of the 1's is untouched and hence still unique. Therefore Γ also satisfies Equation 4.2 for $h = *$.*

If Γ satisfies Equation 4.2 for $h = 0$, then $\overline{\Gamma} = \{\overline{\gamma} \mid \gamma \in \Gamma\}$ satisfies Equation 4.2 for $h = 1$, since $\overline{0} = 1$. For $\gamma' \in \Gamma\&$, if $\mathcal{Z}(\gamma') \in E^{h*}(\Gamma\&, 0)$, the first half of $\mathcal{Z}(\gamma')$ will be unique for unique coalitions, therefore the entire $\mathcal{Z}(\gamma')$ will be unique for unique coalitions. On the other hand, if $\mathcal{Z}(\gamma') \in E^{h*}(\Gamma\&, 1)$, then the second half of $\mathcal{Z}(\gamma')$ will be unique for unique coalitions, therefore the entire $\mathcal{Z}(\gamma')$ will be unique for unique coalitions. Finally, if $\mathcal{Z}(\gamma') \in E^{h*}(\Gamma\&, *)$, the same argument applies, using the first part of this proof.

□

Lemma 2 gives a technique for extending existing binary codes that only satisfy Equation 4.2 for one fixed h. For example, the code in [4, 36] can be extended to satisfy Equation 4.2 for $h = *$ and $h = 1$. The length is only increased by a factor of 2, which does not affect the asymptotic upper-bound length. For example the code in [4, 36] has length $O(c\sqrt{M})$. Applying Lemma 2, the length is $O(2c\sqrt{M}) = O(c\sqrt{M})$. Next, another code that is robust against the hard extended attack for fixed symbol $h = 1$ is presented.

In [4, 36], the *trivial code* against AND attacks of maximum coalition size M, where $|\Gamma| = M$, is presented. Here the code will be called the *identity code*, and it satisfies Equation 4.2 for $h = 1$.

Definition 19 (Identity Code) *Let I_M be a $M \times M$ identity matrix. Then the rows in I_M make up the codewords in Γ, and Γ is called the identity codebook or code.*

Lemma 3 *The identity code satisfies Equation 4.2 for $h = 1$.*

Proof 3 *Suppose any two different arbitrary sized coalitions $\Gamma', \Gamma'' \subset \Gamma$ are chosen. Since $\Gamma' \neq \Gamma''$, $\exists \gamma^i \in \Gamma'$ such that $\gamma^i \notin \Gamma''$ or $\exists \gamma^i \in \Gamma''$ such that $\gamma^i \notin \Gamma'$. Recalling that every codeword in Γ has exactly one 1 at a unique position, let $\gamma^i_k = 1$; that is, the k^{th} bit is the unique 1 in γ^i. Then $\mathcal{Z}(\Gamma')_k = 1$ but $\mathcal{Z}(\Gamma'')_k = 0$ or $\mathcal{Z}(\Gamma'')_k = 1$ but $\mathcal{Z}(\Gamma')_k = 0$. Therefore $\mathcal{Z}(\Gamma') \neq \mathcal{Z}(\Gamma'')$.*

□

The identity code has codeword length $l = O(M)$, however it is robust against coalition sizes of up to M, whereas the $l = O(c\sqrt{M})$ codes in [4, 36] are robust against coalition sizes of $c \leq M$.

Corollary 1 *A codebook Γ with M codewords from the rows of the matrix $[I_M \bar{I}_M]$ satisfies Equation 4.2 for the envelope $E^{h*}(\Gamma, 0) \cup E^{h*}(\Gamma, 1) \cup E^{h*}(\Gamma, *)$.*

Corollary 1 follows from Lemma 2 and 3.

B. Step 2: The Inner Error-Detecting Code

The goal now is to make the code in Lemma 1 even more robust, in particular to the random attack that conforms to the Marking Assumption. The main idea behind achieving this level of robustness is to introduce error-detecting codes, which will be used as an inner code.

Theorem 1 *Given a codebook $\Gamma \subseteq \{0,1\}^l$ of cardinality M that is c-collusion-resistant with respect to the envelope $E^{h*}(\Gamma,0) \cup E^{h*}(\Gamma,1) \cup E^{h*}(\Gamma,*)$, a codebook $\Gamma(C)$ can be created from Γ using error-detection codes C in place of the binary alphabet, such that Γ is the outer code, and C is the inner code, to achieve:*

(1) Robustness against random attacks that conform to the Marking Assumption, such that a detectable bit behaves as though it were being transmitted across the binary random channel in Figure 12 with $p_1 + p_2 + p_3 = 1$.

(2) The probability of error ϵ can be controlled by the error-detecting code.

Proof 4 *Let $C = \{c^0, c^1\}$ be a binary error-detecting code. For each codeword in Γ, map $0 \mapsto c^0$ and $1 \mapsto c^1$. The attack strategy under the envelopes $E^{h*}(\Gamma^e, 0)$ and $E^{h*}(\Gamma^e, 1)$ is taken care of by the fact that Γ is originally immune to such attacks, so assume the attacker randomly chooses 0, 1, or the erasure symbol $*$ with non-zero probability for detectable bits.*

The tracing algorithm will act on the error-detecting blocks in the word $\eta \in E^*(\Gamma^e)$. Each block that is detected to be in error will be replaced by $*$, while blocks that are not in error are mapped back, $c^0 \mapsto 0$ and $c^1 \mapsto 1$. This creates a word $\hat{\eta} \in E^{h*}(\Gamma, *)$, therefore a unique coalition can be identified.

This random attack, acts as random noise on the error-detecting blocks. Suppose C has a probability of failing to detect an error given by ϵ'. Then in the most fatal

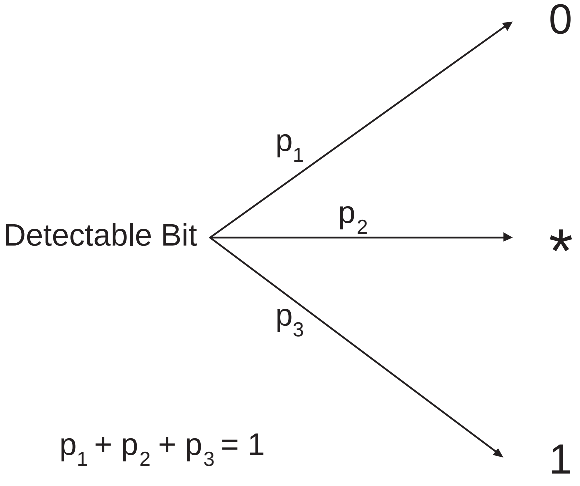


Fig. 12. The Binary Random Channel

case, missing one undetected error will cause $\hat{\eta} \notin E^{h*}(\Gamma, *)$*, and hence* $\hat{\eta}$ *cannot be used to trace the pirates. The probability of being able to trace (i.e.* $\hat{\eta} \in E^{h*}(\Gamma, *)$*) in this case would be* $(1 - \epsilon')^l$*. Hence the probability of error (i.e.* $\hat{\eta} \notin E^{h*}(\Gamma, *)$*) of the entire scheme is bounded by* $\epsilon \leq 1 - (1 - \epsilon')^l$*.*

□

Theorem 1 and its proof provide a technique for designing fingerprinting codes that are resilient against the attack in Figure 12. Usually the pirate will set $p_2 = 0$, degenerating Figure 12 into the binary symmetric channel, because the erasure symbol $*$ incriminates the pirates more easily.

The subsections below summarize the construction and tracing algorithm.

1. Construction

(1) Design a code G that is c-collusion-resistant to attacks that have envelopes of either $E^{h*}(G, 0)$ or $E^{h*}(G, 1)$.

(2) Create the code $\Gamma = G\&$ as outlined in Lemma 2.

(3) Design an error correcting code $C = \{c^0, c^1\}$.

(4) Create the concatenated code (see Definition 16 as well as proof of Theorem 1) with $\Gamma = G\&$ as the outer code, and C as the inner code.

2. Tracing Algorithm

Suppose the attacked word is η.

(1) Break η up into non-overlapping blocks $\{y_1 y_2 \cdots y_l\}$, such that the length of y_i is equal to the length of c^0 or c^1.

(2) For each y_i, use the error-detecting algorithm of C to determine if y_i is in error. If y_i is in error, let $\hat{\eta}_i = *$, otherwise let $\hat{\eta}_i = 0$ if $y_i = c^0$, and $\hat{\eta}_i = 1$ if $y_i = c^1$.

(3) If $\hat{\eta} = \hat{\eta}_1 \hat{\eta}_2 \cdots \hat{\eta}_l \in E^{h*}(\Gamma, *)$, then $\hat{\eta}$ uniquely identifies all pirates.

3. Example of Concatenating an Outer and an Inner Code

In this section, an example of an outer identity code and an inner binary repetition code is provided to illustrate the construction given above. The binary repetition code of length n is given by Definition 20.

Definition 20 (Binary Repetition Code) $C = \left\{ \underbrace{00\cdots0}_{n}, \underbrace{11\cdots1}_{n} \right\}$ *is a* $(n, 1)$ *linear code called the binary repetition code of length* n.

Corollary 2 gives a probability of error bound.

Corollary 2 *Let* W *be any binary* c*-collusion-resistant codebook with codewords of length* l *that is immune to attacks with the attack envelope* $E^{h*}(W, 0) \cup E^{h*}(W, 1) \cup$

$E^{h*}(W, *)$, and let V be the binary repetition code of size n. Construct the fingerprinting code with outer code W, inner binary repetition code V. Suppose the attacker randomly chooses 0, 1, or $*$ when the bit is detectable. Then the probability of error is bounded by Equation 4.3.

$$\begin{aligned} \epsilon &\leq 1 - \left(1 - \frac{1}{2^n}\right)^l \\ &\approx \frac{l}{2^n} \end{aligned} \tag{4.3}$$

Proof 5 *Corollary 2 follows from Theorem 1 by noting that $\epsilon' = \frac{1}{2^n}$ for the binary repetition code. For small $\frac{1}{2^n}$, applying a binomial expansion on the term $\left(1 - \frac{1}{2^n}\right)^l$, and keeping the first two terms, results in the approximation $\frac{l}{2^n}$.*

As an example of using Corollary 2, the codebook consisting of rows from $[I_M, \bar{I}_M]$ from Corollary 1 is used as the outer code, so then length of the outer code is $2M$. The binary repetition code of size n is used as the inner code, so the inner code is of length n. In addition, $\epsilon \leq \frac{2M}{2^n}$, so $n \leq \log_2\left(\frac{2M}{\epsilon}\right)$. Therefore the total length of codewords of this concatenated code is $l = 2Mn \leq 2M\log_2\left(\frac{2M}{\epsilon}\right) = O(M\log_2\left(\frac{M}{\epsilon}\right))$.

C. Step 3: Mixing with Traceability Codes

As mentioned earlier, traceability codes are only resilient against the narrow-sense collusion attack. In this section, a technique is presented that merges traceability codes with the codes introduced in the previous section, allowing the combined code to be resilient to the extended narrow-sense attack.

From [30], traceability codes are defined in Definition 21.

Definition 21 (Traceability Codes) *A code Γ is a c-TA (traceability) code if for* *all coalitions* $\Gamma' \subset \Gamma$ *with* $|\Gamma'| \leq c$*, if* $\eta \in e(\Gamma')$*, then* $\exists \gamma \in \Gamma'$ *such that*

$$|\{i \mid \gamma_i = \eta_i\}| > |\{i \mid z_i = \eta_i\}|$$

$\forall z \in \Gamma \setminus \Gamma'$.

Definition 21 states that whenever a coalition of pirates colludes to produce a word η, there will be a codeword from the coalition that is closer to η (in the Hamming distance) than any codeword not part of the coalition. From this definition, it can be seen that the tracing algorithm is based on finding the closest codeword to the attacked word η in the Hamming distance sense. What is interesting to note, is that the code developed in the previous section has the opposite effect; any erasure of detectable bits for the code developed in the previous section will strongly incriminate the pirates, whereas in traceability codes, erasure is not tolerated, because erasure causes the attacked word to be further in Hamming distance from the codewords owned by the coalition, thus defeating the c-TA tracing algorithm. This insight suggests that there is some way to merge the two codes, thus allowing the resulting code to be robust against the extended narrow-sense attack.[1]

In [30], the following theorem is given that gives the tracing algorithm as well as the required length of c-TA codes.

Theorem 2 *Let* Γ *be a code with codewords that are of length* n*,* c *is a positive integer, and the minimum distance* d *of* C *satisfies* $d > n - \frac{n}{c^2}$*. Then*

(1) C *is a c-TA code;*

(2) if $\Gamma' \subset \Gamma$ *with* $|\Gamma'| \leq c$*, and* $\eta \in e(\Gamma')$*, then*

[1] In [10], c-TA codes are extended to allow for a fixed number of erasures. However the number of erasures allowed is not enough to make the code robust against the extended narrow-sense attack as described in Appendix B.

(i) $\exists\gamma \in \Gamma'$*, such that* $Hamming(\eta, \gamma) \leq n - \frac{n}{c}$*, and*

(ii) $\forall z \in \Gamma$*, if* $Hamming(\eta, z) \leq n - \frac{n}{c}$*, then* $z \in \Gamma'$*.*

Theorem 2 also implies that there exists linear codes with length

$$n \geq c^2 \log_2(q)(\log_q(M) - 1) = c^2 \log_2(M) - c^2 \log_2(q)$$

bits, where where c is the maximum size of the coalition, M is the total number of users supported, and q is the size of the alphabet (i.e. $q = |\Sigma|$). Minimum Distance Separable (MDS) codes such as Reed-Solomon codes are often used in c-TA codes [30, 31, 10], resulting in a length of exactly $n = c^2 \log_2(M)$ bits.

Now to extend the c-TA MDS code to tolerate erasures, adjoin to the c-TA MDS code, the identity code I_M (Definition 19), and repeat each column in the identity code $d \times c^2 log_2(M)$ times. Then randomly permute the columns of the code identity and c-TA code, intermixing them, keeping the permutation secret from the users so that the users do not know the locations of the identity code and the c-TA code. When the pirates choose to erase one detectable bit, the probability that this bit belongs to the c-TA code is bounded by Equation 4.5.

$$\begin{aligned} \epsilon &\leq \frac{c^2 \log_2(M)}{c^2 \log_2(M) + 2dc^2 log_2(M)} && (4.4) \\ &= \frac{1}{1 + 2d} && (4.5) \end{aligned}$$

Although the length of the combined c-TA and identity code is

$$c^2 \log_2(M) + Mdc^2 log_2(M),$$

the denominator in Equation 4.4 is only $c^2 \log_2(M) + 2dc^2 log_2(M)$, because in the worst case scenario, when only 2 pirates collude, only 2 bits from the identity code are detectable. In fact, as more pirates collude, more bits from the identity code are detectable, and hence it is even more difficult for the pirates to erase a detectable bit from the c-TA code. In addition, if d is large enough, the more detectable bits the pirates erase, the more likely these erased bits will fall into the identity code, which will then incriminate the pirates as discussed in the previous section.

For any fixed ϵ, $d \leq \frac{1}{2}\left(\frac{1}{\epsilon} - 1\right)$ by solving for d in Equation 4.5. Therefore the entire length of the joint c-TA and identity code is given by Equation 4.6.

$$\begin{aligned} l &\leq c^2 \log_2(M)\left(1 + \frac{M}{2}\left(\frac{1}{\epsilon} - 1\right)\right) & (4.6) \\ &= O\left(\frac{c^2}{\epsilon} M \log_2(M)\right) & (4.7) \end{aligned}$$

This code is not asymptotically better than the code in [3], whose length is given by Equation 3.5, when all the parameters c, M, and ϵ are considered as independent variables. However, it will be shown in the next section that when M is a function of c, such that c is some fixed percentage of M, and ϵ is held constant, this code has much shorter codeword length.

D. The Malicious Distributor

The fingerprinting research has been active for more than 10 years, and as mentioned earlier, there are several different criteria used in comparing the performance of codes. In this section, the performance criterion used in this work will first be motivated, and then it will be shown that the proposed code excels under this criterion.

In the classical collusion scenario, a small coalition of buyers with legitimate copies compare their fingerprinted copies and try to detect and remove their fingerprints. This classical collusion scenario can either consist of a small group of people meeting one another in person, on the Internet, or even one person purchasing a small number of copies. In this scenario, it makes sense that the fingerprinting codes are designed for a fixed small number c, where c is the maximum allowed number of colluders. Indeed, the fingerprinting schemes reviewed in this work have been designed for such a scenario, as their codeword lengths vary exponentially with c [33, 8, 19], or are exclusively designed for a small coalition size such as 2 or 3 [9, 6, 27, 19].

The second collusion scenario that has received little attention is that of a malicious distributor. Suppose the broadcasting framework is such that there exists several different distributors (such as different cable or satellite service providers). Each distributor receives some non-negligible percentage of the total number of fingerprinted media. A malicious distributor or perhaps a malicious employee who obtains a large number of fingerprinted media and applies collusion attacks, will defeat the tracing algorithm when these codes have a fixed c that is small. Upon removing the fingerprints via large-scale collusion, the malicious distributor might then illegally sell the untraceable media to generate additional income.

In general, the codeword length is a function of (M, c, ϵ), which represent the total number of users supported, the maximum coalition size, and the probability of error respectively. Under the malicious distributor scenario, (M, c) can be collapsed into one parameter, because c is a fixed percentage of M, and therefore M is a function of c. For example, if c is 10% of M (i.e. $M = 10c$) then $10c$ can replace the parameter M. In addition, the parameter ϵ is usually fixed and user-defined [3]. Therefore, under the malicious distributor scenario, the codeword length is only a function of the variable c, which itself is a function of the number of distributors. The interested reader is referred to Appendix C, which discusses how modulation and watermarking can be used to prevent the exploitation of the structure of the code in collusion attacks.

The following section will provide empirical results that demonstrate the superiority of the codeword length of the proposed code under the malicious distributor scenario.

E. Comparison of Codes

Under the malicious distributor scenario, let $M = \nu c$ for some constant ν, and suppose ϵ is fixed, say to 10^{-5}. Then the codeword length in [3] is $O(c^4 \log_2(c))$, while the codeword length of the proposed code is $O(c^3 \log_2(c))$. Therefore under the malicious distributor scenario, the proposed code is superior in codeword length. Figure 13 demonstrates that the length of the codewords for the proposed code is shorter than the length of the codewords for the code in [3] when c is some percentage of M, and ϵ is fixed at 10^{-5}.

The other codes presented in this thesis are not plotted, because their codeword lengths grow exponentially with c. A summary of the codes is provided in Table I, outlining the strengths and weaknesses of each code.

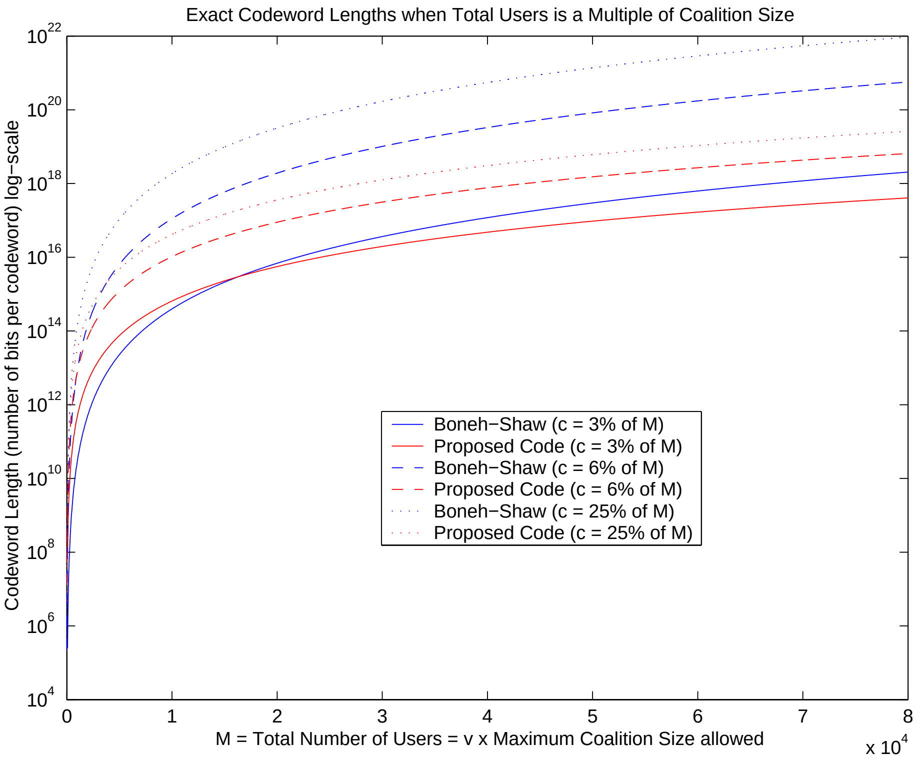


Fig. 13. Codeword Lengths for the Boneh-Shaw Code vs. the Proposed Code When the Coalition Size Is a Percentage of the Total Number of Users and ϵ Is Fixed at 10^{-5}

Table I. Comparison of Popular Fingerprinting Codes

Method	**Robustness**	**Coalition Size**	**Length in terms of coalition size c (other parameters constant)**	**Number of Pirates Identified**	**Probability of Error**	**Major Drawback**
Projective Geometric	Marking Assumption with erasure	Small c	Exponential in c	All	User-defined parameter	Small codebooks (hence small coalition sizes).
BIBD	Binary AND	At most codeword length over the square root of the total number of users	Approximately square of c	All	0	Only robust to binary AND operator.
Boneh and Shaw concatenated code	Marking Assumption with erasure	Large c	Approximately 4^{th} power in c	One	User-defined parameter	For fixed probability of error, and number of users as a function of c, length is longer than proposed novel code.
Separating concatenated code	Marking Assumption with erasure	Small c	Exponential for large c	One	At least decreasing exponentially in codeword length	Not appropriate for large coalitions.
Novel Code	Marking Assumption with erasure	Large c	Approximately cube of c	At least one	User-defined parameter	For varying probability of error, number of users, and coalition size, length is longer than Boneh-Shaw

CHAPTER V

JOINT SOURCE FINGERPRINTING - A NEW PARADIGM FOR MULTIMEDIA FINGERPRINTING

This chapter introduces a new paradigm for multimedia fingerprinting. The shortcomings of the traditional paradigm for multimedia fingerprinting are briefly reviewed. Motivated by a need to address these shortcomings, a new paradigm termed *Joint Source Fingerprinting* is introduced. Insights into the new JSF paradigm are presented by comparing the new paradigm to existing research. An algorithm for video fingerprinting is then presented, along with simulation results to verify the theoretical and empirical observations.

A. Shortcomings of Traditional Fingerprinting

In the traditional multi-step fingerprinting paradigm, a codebook (set of codewords) with collusion-resistant properties is designed independently of the media into which the codewords are to be embedded. The codewords are then modulated and embedded into the host media via watermarking techniques, resulting in uniquely fingerprinted media. A pirate can then obtain several uniquely fingerprinted copies with the intention of either removing the fingerprints, or scrambling the fingerprints so as to frame innocent users.

The shortcomings of this fingerprinting approach are:

(1) The fingerprint codewords are designed independently of the media, and only merged with the media via a watermarking technique. Therefore this method of "fingerprinting using codebook and watermarking" is not specialized for multimedia. In most cases, the codebook attacks are not representative of the

multimedia collusion attacks, nor the single-user attacks discussed in Chapter II (i.e. the Marking Assumption may be ignored by pirates).

(2) The codeword length required to support a large number of users may be too long, hence the entire codeword may not "fit" into the media using watermarking techniques.

(3) As more pirates participate in collusion attacks, the fingerprints become more vulnerable to removal, yet the media itself does not suffer from visual degradation.

(4) The uniqueness requirement of fingerprint codewords, conflicts with the requirements of statistical invisibility [42].

The first shortcoming is prevalent in the fingerprinting literature. For example, attackers may not conform to the Marking Assumption by applying any sort of single-user attack at their discretion.

The second shortcoming is based on the idea that every media has a watermarking capacity, such that if the watermark to be embedded is larger than this capacity, the resulting media either experiences visual distortion, or the watermark cannot be reliably extracted [43, 44].

The third shortcoming states that a multimedia collusion attack does not affect the visual quality of multimedia such as images or videos. This statement is verified in the simulation results later in this chapter. However, a collusion attack that nullifies the fingerprints should also be required to result in visually degraded multimedia; that is, colluders should be punished with visually degraded multimedia whenever collusion attacks are employed. The spirit of punishing colluders with degraded media is also found in [45]. In [45], images are fingerprinted by applying a random geometric warp (rotation and translation). The geometric warp itself does not visually degrade the image. However when uniquely fingerprinted copies are averaged, the resulting image appears blurry, and hence it is of little commercial value. In essence, an average attack is punished with severe visual degradation. The shortcoming of [45] is that it is easy to remove the fingerprint via single-user attacks (i.e. further rotating and translating of the image will confuse the tracing algorithm). As will be shown, the JSF adopts this idea, but improves upon it by punishing single-user attacks as well as collusion attacks.

Finally, [42] equates the inability to detect a watermark to that of statistical invisibility, which is satisfied if and only if the correlation between two frames/images is equal to the correlation between two watermarks that are to be embedded into the two frames/images. When dealing with multimedia fingerprinting using codebook and watermarking, the two images are identical, and hence to achieve statistical invisibility, the two fingerprints have to be identical as well. This contradicts the uniqueness requirements of fingerprints, and at the same time suggests that fingerprinting using codebook and watermarking may not be appropriate for multimedia fingerprinting.

B. Joint Source Fingerprinting

This brief introduction gives the general idea behind the JSF paradigm. In the next section, a mathematical description is provided to solidify the concepts presented here.

The JSF paradigm integrates codebook design with the multimedia itself. That is, the multimedia is used as the fundamental alphabet (i.e. Σ) of the codebook. This approach is a significant departure from the traditional approach, and paves the way for overcoming the first shortcoming described above. The idea is to separate the multimedia into two classes, termed the *semantic class*, and the *feature class*. These two classes can contain elements from any transform of the multimedia, such as the Discrete Cosine Transform (DCT), or the Discrete Wavelet Transform (DWT). Each resulting fingerprinted multimedia will include the semantic class, but different subsets of the feature class. These different subsets of the feature class are called the *fingerprints*.

The semantic class is a coarse representation of the multimedia that gives enough information to understand the semantic content of the multimedia, however it lacks detail, and hence has no commercial value. Examples of semantic classes for images are the edges of an image, the low-pass filtered image, half an image (i.e. crop the bottom half of the image, but keep the top half), or any other coarse representation of an image.

The idea is that colluders can distinguish the fingerprints amongst different copies, however completely removing the fingerprints will result in the semantic class, which has no commercial value. Hence this is a direction towards partially overcoming the third shortcoming in the traditional approach.

Finally the fingerprints should be difficult to combine, and any computationally simple combination of fingerprints will either still incriminate at least one of the colluders, or result in a visually degraded copy. For example, just mixing fingerprints from different users will result in the identification of all colluders whose fingerprints are part of this mix.

1. Mathematical Description of the Joint Source Fingerprinting Paradigm

The JSF paradigm is now defined mathematically. As a first step, visual entropy is defined in order to account for the Human Visual System (HVS), as this is the fundamental building block in image and video compression, watermarking, as well as the JSF paradigm. For example, the HVS ignores many details in photographic stills, and these ignored details are partially removed in compression algorithms, or partially used to hide watermarks[1].

Definition 22 (Visual Entropy) *Let C be a random vector representing some multimedia content. Let p_C be the probability mass function of C. Then the visual entropy of C is defined as*

$$H_V(C) = \sum_x p_C(x) \log_2 \frac{1}{p_C(x)} - V(C) \tag{5.1}$$

where $V(C)$ measures the amount of information in C that is redundant to the HVS - that is the HVS cannot perceive this information.

In Equation 5.1, $V(C)$ is either determined experimentally, or derived from well-known HVS models [46]. This redundant information is then subtracted from the standard definition of entropy to give rise to visual entropy. Therefore visual entropy can also be written as

$$H_V(C) = H(C) - V(C) \tag{5.2}$$

In Equation 5.2, $H(C)$ is the standard definition of entropy.

Notation 1 (Equality of Visual Entropies) *$H_V(C_1) = H_V(C_2)$ if and only if the quantities $H_V(C_1)$ and $H_V(C_2)$ are* equal*, and C_1 is visually* equal *to C_2. All other relational operators are defined so that the word* <u>equal</u> *in the above statement, is replaced with the respective relational operator.*

Since visual entropy only gives a number as to how much information the HVS absorbs, Notation 1 is required to further establish that two multimedia *look* the same

[1] This section focuses on images and videos, however the same principles can be applied for audio media, as the Human Audio System also ignores details, such as frequencies outside the human auditory system.

or *look* different.

It should also be noted that visual entropy is not additive. For example, in general

$$H_V(C_1 \cup C_2) \neq H_V(C_1) + H_V(C_2). \tag{5.3}$$

In addition,

$$H_V(C_1 \cup C_2) \nleq H_V(C_1) + H_V(C_2) \tag{5.4}$$

and

$$H_V(C_1 \cup C_2) \ngeq H_V(C_1) + H_V(C_2). \tag{5.5}$$

This is due the subjectiveness of $V(C)$ in Equation 22. However Equations 5.6 and 5.7 are true if C_1 and C_2 are elements from the same image or video.

$$\begin{aligned} H_V(C_1 \cup C_2) &\geq H_V(C_1) && (5.6) \\ H_V(C_1 \cup C_2) &\geq H_V(C_2) && (5.7) \end{aligned}$$

For example, if $C = C_1 \cup C_2 \cup \ldots$, such that C_1 and C_2 are frames from the video C, then adding additional frames from C will only improve the visual quality, but never degrade the visual quality.

The semantic and feature classes, as described in the above section, are now formally introduced.

Definition 23 (Semantic Class) *Let Ξ be the the semantic class for some multimedia content C, such that $\Xi \subset T(C)$, where $T(\cdot)$ is some transform. The following relationship is true*

$$H_V(\Xi) \ll H_V(C) \tag{5.8}$$

In addition, $T^{-1}(\Xi)$ should provide some semantic information about C.

Definition 24 (Semantic-Feature Representation) (Ξ, Φ) *is called the Semantic-*

Feature Representation of C, when Ξ is as defined in Definition 23, and $\Phi = T(C)\backslash\Xi$ is the feature class. The notation is written as $C \sim (\Xi, \Phi)$.

It should be noted that in Definition 24, (Ξ, Φ) contains the same visual information as C, which implies $H_V(\Xi, \Phi) = H_V(C)$. Fingerprints were described as subsets of the feature class in the previous section, however Definition 25 gives an additional requirement.

Definition 25 (Fingerprint) *A fingerprint $\Phi_i \subseteq g_i(\Phi)$ obeys the following relationship*

$$\forall i \;\; H_V(\Xi) \ll H_V(\Xi, \Phi_i) \approx H_V(C) \tag{5.9}$$

Definition 25 requires that the combination of the semantic class along with a fingerprint is visually similar to the original image or video. Also, in this thesis, $g_i(\cdot)$ is the identity function (i.e. $g_i(x) = x$) for all i.

The Mixed Semantic-Feature Representation is now defined to show how the JSF paradigm and the watermarking paradigm are related. However, this definition can be skipped for the purposes of understanding the JSF paradigm alone.

Definition 26 (Mixed Semantic-Feature Representation) *(Ξ, Φ) is called the Mixed Semantic-Feature Representation of C when $\Xi \subset T(C, I)$, where $T(\cdot, \cdot)$ is a transform that takes I as side information, and $\Phi = U(C, J)$, where $U(\cdot, \cdot)$ is a transform that takes J as side information. Ξ obeys the relationship in Definition 23, and also provides semantic information about C. In addition $|H_V(C) - H_V(\Xi, \Phi)| < \epsilon$ for small $\epsilon \geq 0$. The notation is written as $C \asymp (\Xi, \Phi)$*

In the Mixed Semantic-Feature Representation, side information is added to define both the semantic class as well as the feature class. The transforms applied to obtain the two classes may also differ. The Semantic-Feature Representation is a specific case of the Mixed Semantic-Feature Representation, when $I = J = \varnothing$ contain no information, and $T = U$.

Definition 27 (Quasi-JSF Fingerprint) *A fingerprint $\Phi_i \subseteq g_i(\Phi, W)$ obeys the following relationship*

$$\forall i \; H_V(\Xi) \ll H_V(\Xi, \Phi_i) \approx H_V(C) \tag{5.10}$$

Here $g_i(\cdot,\cdot)$ uses additional information W, independent of C.

Recall that an earlier statement established that the multimedia itself is used as the fundamental alphabet. This statement is now mathematically elaborated by first noting that the JSF fingerprints can also be described as a binary fingerprint code. Let the elements in $\Phi = \{\phi_j\}$ be enumerated, i.e. $(\phi_1, \phi_2, \phi_3, \ldots)$. Using Definition 25 with g_i as the identity function, every fingerprint Φ_i can be described as a binary codeword; if $\phi_k \in \Phi_i$ then position k in $(\phi_1, \phi_2, \phi_3, \ldots)$ is a 1, otherwise it is a 0. For example, assume that $|\Phi| = 4$, $\phi_2, \phi_4 \in \Phi_i$, and $\phi_1, \phi_3 \notin \Phi_i$, then the codeword for Φ_i is $\gamma^i = \gamma^i_1\gamma^i_2\gamma^i_3\gamma^i_4 = 0101$.

Definition 28 (JSF Fingerprint Codeword) *Let the elements in Φ be enumerated as $(\phi_1, \phi_2, \phi_3, \ldots, \phi_N)$. The binary codeword for a fingerprint $\Phi_i \subset \Phi$, is $\gamma^i = \gamma^i_1\gamma^i_2 \cdots \gamma^i_N$, where*

$$\gamma^i_j = \begin{cases} 1 & \textit{if} \;\; \phi_j \in \Phi_i \\ 0 & \textit{if} \;\; \phi_j \notin \Phi_i \end{cases}$$

The concept of collusion-resistance that also overcomes Shortcoming 3 in Section A of this chapter, is now defined.

Definition 29 (Collusion-Resistant Fingerprint) *A set of fingerprints, $\{\Phi_i\}$, is said to be collusion-resistant if it is computationally difficult to create some $\widetilde{\Phi} =$*

$\mathcal{Z}(\{\Phi_i\})$, where $\mathcal{Z}$ is a collusion attack, and $\forall i\ \widetilde{\Phi} \cap \Phi_i = \emptyset$, such that

$$H_V(\Xi, \widetilde{\Phi}) \approx H_V(C)$$

This last property states that it is computationally cumbersome to create a new fingerprint that does not contain any of the original fingerprints, and at the same time has no visual distortion.

A JSF algorithm is *optimal* if all other algorithms produce fewer (or a equal number of) fingerprints. To formally define optimality, the fingerprinting capacity is introduced.

Definition 30 (Fingerprinting Capacity) *The fingerprinting capacity $\mathcal{C}$, of a media C, is defined to be the maximum number of collusion-resistant fingerprints $\{\Phi_i\}$ that can be generated from C, for all semantic-feature representations (Ξ, Φ):*

$$\mathcal{C} = \max_{(\Xi,\Phi),\Phi_i \subset \Phi} |\{\Phi_i\}| \tag{5.11}$$

Definition 31 (Optimal JSF Algorithm) *A JSF algorithm is said to be optimal if and only if the algorithm achieves the fingerprinting capacity.*

Note that Definition 30 provides no tractable method of finding the fingerprinting capacity, and determination of this quantity is an open question.

A summary of the JSF design is give below:

(1) Choose a transform $T(\cdot)$

(2) Separate the multimedia C into its Semantic-Feature Representation, (Ξ, Φ).

(3) Choose a transform $g_i(\cdot)$ (possibly the identity function, i.e. $g_i(x) = x$).

(4) Separate $g_i(\Phi)$ into collusion-resistant fingerprints $\{\Phi_i\}$.

These design steps integrate the multimedia with the fingerprint design, creating a codebook of collusion-resistant fingerprints ($\{\Phi_i\}$ in the form given by Definition 28) *from the source itself*, hence the term Joint Source Fingerprinting.

2. Detection of Fingerprints

Some general detection techniques for the JSF paradigm are now presented. The actual transform and fingerprint selection process will determine the exact, simplified detector. Given a possibly corrupted copy of some fingerprinted multimedia $(\Xi, \widetilde{\Phi})$, the idea is to detect traces of the original fingerprints $\{\Phi_i\}$ in $\widetilde{\Phi}$, and list any possible suspects. Since the methods described for tracing pirates are traditionally known as detectors, the tracing algorithm will also be called a *detector*. This section describes both hard-decision and soft-decision detectors.

A detector takes a sequence of corrupted fingerprints $(\tilde{\phi}_i)$, and passes each $\tilde{\phi}_i$ through a function $h(\cdot)$. Recall that in the hard-decision decoder, $h(\tilde{\phi}_i)$ is then compared to a threshold τ_1, and mapped to 1 if $h(\tilde{\phi}_i) > \tau_1$, or mapped to 0 if $h(\tilde{\phi}_i) < \tau_1$. After processing all $\tilde{\phi}_i$ in this manner, a word $\tilde{\gamma}$, consisting of 1's and 0's is created. This binary word $\tilde{\gamma}$ is then compared to all pristine binary codewords $\{\gamma^i\}$ representing the pristine fingerprints as defined in Definition 28. The word in $\{\gamma^i\}$ with the minimum Hamming distance to $\tilde{\gamma}$ is selected.

In the soft-decision detector, $h(\tilde{\phi}_i)$ is not hard-limited; this means it is not compared to a threshold τ_1, and not mapped to 1 or 0. There are two popular soft-decision decoders: the maximum likelihood detector and maximum a posteriori detector. A *maximum likelihood* (ML) detector is given by Equation 5.12.

$$\text{User } i \text{ is a suspect} = \arg\max_i Pr(\ \widetilde{\Phi} \mid \Phi_i\ ) \tag{5.12}$$

Similarly the *maximum a posteriori* (MAP) detector is given in Equation 5.13.

$$\text{User } i \text{ is a suspect} = \arg\max_i Pr(\ \Phi_i \mid \widetilde{\Phi}\ ) \tag{5.13}$$

The hard-decision detector is easier to implement, and also costs less in terms of computational power and memory. However, the hard-decision detector is usually suboptimal. The soft-decision detectors are better than the hard-decision detectors in terms of lower probability of error, if the conditional probabilities are known.

3. Immunity Against Attacks

As mentioned earlier, there are two main types of attacks that colluders can apply: an estimation attack, and a scrambling attack. In the JSF paradigm, the estimation attack is not effective if the coalition size is equal to the number of fingerprinted copies. For example, colluders can combine all their fingerprints to produce $(\Xi, \widetilde{\Phi})$, the closest estimate to $C \sim (\Xi, \Phi)$. However, given $\Phi_i \subset \Phi$, $\widetilde{\Phi} = \bigcup_i \Phi_i$ *is* the closest estimate of C, and hence all the colluders will be detected. On the other hand, removal of all fingerprints will result in only the semantic class $(\Xi, \emptyset)$, and this does not have any commercial value. Recall that in the traditional fingerprinting paradigm, it is possible to remove all fingerprints, resulting in an original unfingerprinted copy. In the JSF paradigm, this has been shown to result in the semantic class.

The second type of attack is a scrambling attack. Such attacks try to scramble the fingerprints in such a way that detection of any one colluder is impossible. In the JSF paradigm, colluders would try to create $\widetilde{\Phi}$ from their sets of fingerprints $\{\Phi_i\}$, such that:

(1) A detector cannot trace the origin of $\widetilde{\Phi}$;

(2) $(\Xi, \widetilde{\Phi}) \approx (\Xi, \Phi)$.

According to the JSF paradigm, it is computationally complex to create a $\widetilde{\Phi}$ such that $H_V(\Xi, \widetilde{\Phi}) \approx H_V(C)$, therefore the scrambling attack is difficult.

In the traditional fingerprinting paradigm, the locations of the fingerprints are unbeknownst to the attacker, however in the JSF paradigm, this is not the case, and the colluders can see exactly where the fingerprints are. The difficulty is in manipulating them in such a way as to create new fingerprints that do not exhibit visual degradation in the resulting media. In addition, attack assumptions, such as the Marking Assumption, are not necessary. Therefore it should be stressed that simply comparing the probability of error in the JSF scheme to other schemes is not enough. Whenever the colluders manage to successfully scramble the fingerprint, visual distortion should be present, and hence a missed detection is more acceptable.

C. Insights and Implications

This section consists of two major parts. The first subsection will compare the JSF paradigm to other similar fields of research. The second subsection will propose possible implementation strategies using the JSF, hence showing its potential.

1. Joint Source Fingerprinting in Relation to Other Fields of Research

This section examines how the JSF paradigm is related to similar research fields. The two similar research fields are multimedia data compression, and digital watermarking.

a. Joint Source Fingerprinting in Relation to Digital Watermarking

In most digital watermarking schemes, the watermark or fingerprint is embedded in the transform domain in the mid to high frequency components, where the watermark is not visible [16].

Suppose $T(\cdot)$ is a frequency transform (such as DCT or DWT), and let Ξ be the lowpass-filtered $T(C)$. Then $\Phi = T(C) \setminus \Xi$ represents the mid to high frequency components of C. Now $\Phi_i = g_i(\Phi, W_i) = \Phi \oplus W_i$, where W_i is the watermark/fingerprint created from a collusion-resistant codebook, and it is added (denoted as $\oplus$) to the mid to high frequency components.

From this analysis, digital watermarking is a specific example of the Quasi-JSF paradigm. Digital watermarking assumes the semantic class to be a low frequency representation of C. In the JSF paradigm, this need not be the case. In fact, a high frequency representation of C gives enough semantic information, as the outline of objects gives enough visual information as to what an object may be [47]. Therefore the JSF paradigm is more general than the traditional fingerprinting paradigm.

b. Joint Source Fingerprinting in Relation to Data Compression for Multimedia

The JSF paradigm makes use of visual entropies as does data compression for multimedia. In multimedia data compression, the goal is also to segregate the multimedia C, into two classes (Ξ', Φ'). Here $H_V(C) \approx H_V(\Xi')$, and Φ' contains the visually redundant information $V(C)$, as defined in Definition 22. Hence Φ' can be discarded, resulting in Ξ', a compressed version of C. A good multimedia data compression scheme tries to segregate C into (Ξ', Φ') in such a way that Φ' is as large as possible, and Ξ' is as small as possible, while still maintaining $H_V(C) \approx H_V(\Xi')$. In the JSF paradigm, the segregation (Ξ, Φ) is performed in such a way so that Ξ gives enough semantic information, but has no commercial value because it is a visually degraded version of C. This leaves Φ with redundant information as well as visually important information that when added to Ξ, produces a closer approximation to C. The relationship between JSF and multimedia data compression is given by Equations 5.14

and 5.15.

$$
\begin{aligned}
H_V(\Xi) &< H_V(\Xi') \approx H_V(C) && (5.14) \\
H_V(\Phi') &< H_V(\Phi) && (5.15)
\end{aligned}
$$

One might envision that (Ξ, Φ) can be arrived at by taking (Ξ', Φ'), and donating elements of Ξ' to Φ', thereby increasing visual information in Φ' (the result is Φ), and reducing visual information in Ξ' (the result is Ξ).

2. Examples of Techniques Using the JSF Paradigm

This section presents some practical as well as abstract examples of fingerprinting techniques and attacks under the JSF paradigm.

a. Multiple Compression Units

In this scheme, suppose there are N different non-ideal (i.e. there are always some visual redundancies left over after compression) multimedia compression algorithms. The multimedia C is processed by each of the N compression algorithms, resulting in N different outputs $\{C_i\}_{i=1}^N$. This technique actually conforms to part of the JSF paradigm. Since the compression algorithms are not ideal, (i.e. there are always some redundancies leftover), each of the $\{C_i\}_{i=1}^N$ will be slightly different. Since $\{C_i\}_{i=1}^N$ are all visually identical to C, they share the same semantic class Ξ . The fingerprints are then defined as $\Phi_i = T(C) \setminus \Xi$. Note however, that the fingerprints may not be collusion-resistant, and the presence or absence of the collusion-resistant property depends on the compression algorithms used. Figure 14 depicts the multiple compressor method.

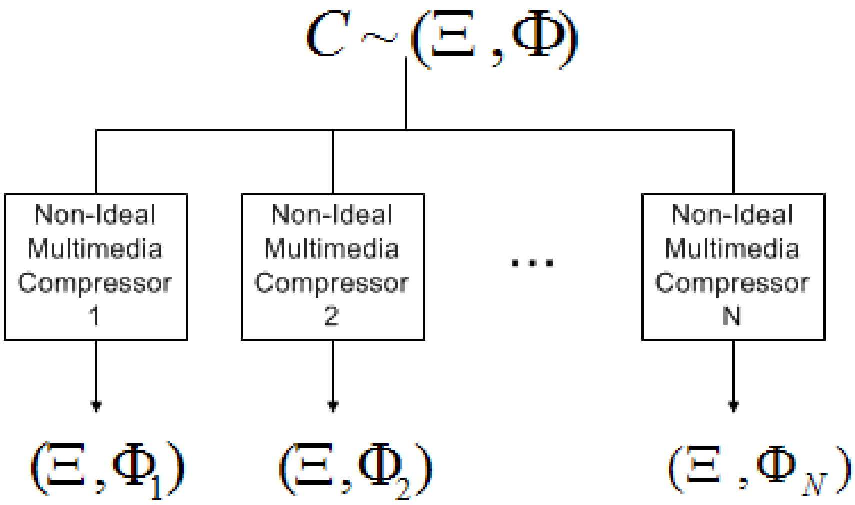

$C \sim (\Xi, \Phi)$
Non-Ideal Multimedia Compressor 1
Non-Ideal Multimedia Compressor 2
...
Non-Ideal Multimedia Compressor N
(Ξ, Φ_1)
(Ξ, Φ_2)
(Ξ, Φ_N)

Fig. 14. Fingerprinted Media from Multiple Compression Units

b. Ideal Compression Attack

Suppose that there exists an ideal compression algorithm $\iota(\cdot)$, that removes all visual redundancies, i.e. $H(\iota(C)) = H_V(\iota(C))$.[2] In this case $\iota((\Xi, \Phi_i)) = \iota((\Xi, \Phi_j))$ for all i, j. Any detection scheme would fail, since any fingerprinted media (Ξ, Φ_i), can result in $\iota((\Xi, \Phi))$. Fortunately ideal compression of multimedia does not exist, since $V(C)$ is experimentally determined, hence absolute removal of visual redundancy in any media is impossible. Any non-ideal compression algorithm used on (Ξ, Φ_i) would result in a different output for different i, since some visual redundancies are not removed.

[2] When the visual entropy is equal to the standard entropy, $V(C) = 0$, implying there are no visual redundancies.

c. Frame-based JSF for Video

Perhaps the simplest and most elegant example of using the JSF paradigm is the frame-based method for video.

First, suppose C is a video with a very high frame rate (i.e. many frames per second). Let the transform $T(\cdot)$ separate the video C into frames, $T(C) = \{c_1, c_2, \ldots c_N\}$. Now let Ξ be a decimated version of $\{c_1, c_2, \ldots c_N\}$.[3] The remaining frames that were decimated now constitute $\Phi = T(C) \setminus \Xi$. Φ can now be separated into M partitions $\{\Phi_i\}$ (i.e. $\Phi = \bigcup_{i=1}^{M} \Phi_i$), such that $\forall i \neq j \;\Phi_i \cap \Phi_j = \emptyset$, and $\forall i = 1, 2, \ldots M \; H_V(\Xi, \Phi_i) \approx H_V(C)$.

Suppose the colluders try to remove the fingerprints. With each frame that they discard, the video becomes increasingly distorted. When all the fingerprints are removed, the colluders are left with a motion-choppy video, which has no commercial value. Now suppose the colluders combine their fingerprints by including all frames from their copies. This would negligibly enhance the video quality, however all the colluders' fingerprints are present in this video.

Finally, suppose the colluders average corresponding frames from their copies. Since each frame differs a little, objects in each frame are shifted a little, therefore averaging of corresponding frames will result in blurry frames [45]. The colluders may try to accurately register each frame (that is align the objects in each corresponding frame), and then average the corresponding frames. This is a computationally complex process, as there is a need to first search for differing objects, and then spatially synchronize them [48]. It is this lack of spatial synchronization that makes attacks more difficult. Figure 15 depicts the lack of spatial synchronization between frames.

[3] Ξ is similar to a video from the Charlie Chaplin era - although crude, the semantics of the film is still understood by the audience.

The circle in the frame on the right is in its original position. The solid circle in the frame on the left is in its new position (original position is faded and dotted in the left frame), and therefore is not synchronized with the circle in its original position. In the traditional fingerprinting paradigm using watermarking, all fingerprinted copies are synchronized, therefore collusion attacks such as averaging do not produce visual distortion. The frame-based method introduced in this section eliminates synchronization, thereby punishing colluders with visual distortion when collusion attacks are applied.

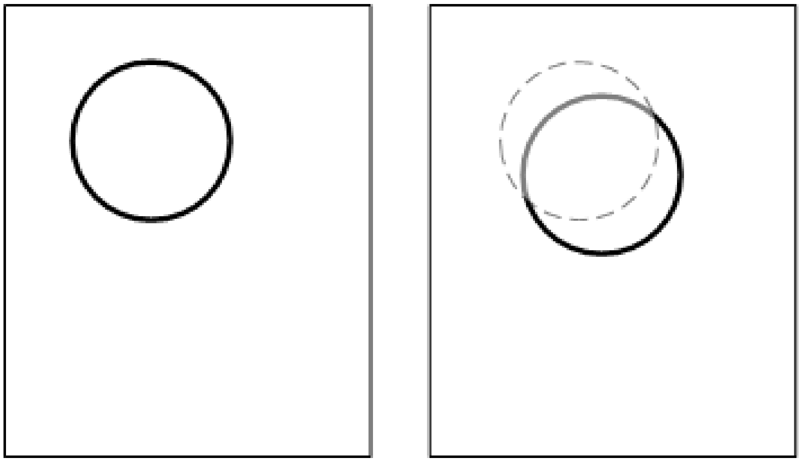


Fig. 15. Spatial Desynchronization of Objects Between Frames

The next section develops the frame-based JSF algorithm for video, and subsequent sections provide simulation results.

D. A Suboptimal JSF Algorithm for Video

In this section, a practical suboptimal JSF algorithm is presented for video sequences. The presentation of this algorithm is divided into the three parts. The first part shows how to derive the semantic class, while the second part shows how the fingerprints are selected. Finally the last part will derive the detector, or tracing algorithm.

The transform $T(\cdot)$ used for video input will consist of breaking the video into frames, which is the frame-based method discussed in Section c of this chapter. To recapitulate, the frame-based method decimates (also known as *downsampling*) frames from a video sequence, resulting in the semantic class. The specific downsampling rate will be derived in the next section. Frames that are not part of the semantic class, i.e. the feature class, are then partitioned so that each partition along with the semantic class will result in a video that is visually similar to the original video.

1. Deriving the Semantic Class

The semantic class should bare some resemblance to the original video sequence, but it should be visually degraded to the point that it has no commercial value. In the frame-based method, the semantic class can be derived by decimating or downsampling frames from the original video sequence. The question is what rate should the downsampler be in order for the downsampled video to have no commercial value? The answer lies in the downsampling theory from digital signal processing theory.

If a discrete-time signal $x(n)$ is downsampled by a factor of D (i.e. the resulting dowsampled signal $x_d(n)$ can be written in terms of the original signal $x(n)$ as $x_d(n) = x(Dn)$), then frequency aliasing will occur if the discrete-time Fourier transform of $x(n)$, denoted as $X(\omega)$, has frequency components in $-\pi \leq \omega < -\frac{\pi}{D}$ or $\frac{\pi}{D} < \omega \leq \pi$. When frequency aliasing occurs, there is no way to recover the original signal $x(n)$ from its downsampled version. This is an attractive feature that can be used in creating the semantic class, because an attacker cannot recover the original video from a frequency aliased version of the original video.

The first step is to find the frequency spectrum of the original video, so that the factor D required for frequency aliasing can be determined. However, a video sequence is represented by a 3-dimensional data structure (space and time), while the downsampling theory is for 1-dimensional signals. The dimension of interest, is the temporal axis, since downsampling is with respect to frames, which encode time, so a video sequence can be transformed into a 1-dimensional signal. First, each frame is partitioned into non-overlapping 8×8 or 16×16 macro-blocks, which is customary in the field of image and video compression research [46]. Let $B_i^n(x, y)$ be a $b \times b$ macro-block where n is the frame number, (x, y) is the pixel coordinate relative to a corner of the macro-block, and i indexes the macro-block within one frame. Each macro block is then averaged, $\frac{1}{b \times b} \sum_{x=1}^{b} \sum_{y=1}^{b} B_i^n(x, y)$, and the averaged results for macro-blocks aligned temporarily will constitute the 1-dimensional signal $x_i(n)$. The cuttoff frequency $\omega_c(i)$ is determined from $X_i(\omega)$, the Fourier transform of $x_i(n)$. This value is defined by Definition 32.

Definition 32 *$\omega_c > 0$ is the ϵ-cutoff frequency of $X(\omega)$ when ω_c is the largest value such that $|X(\omega)| < \epsilon$ when $\omega_c < \omega < 2\pi - \omega_c$.*

Now the downsampling factor D_i for the set of averaged ith macro-blocks given by Equation 5.16 will guarantee frequency aliasing for that set of aligned macro-blocks alone.

$$D_i = \left\lceil \frac{\pi}{\omega_c(i)} \right\rceil + 1 \tag{5.16}$$

Here $\lceil n \rceil$ is the ceiling function that returns the first integer greater than or equal to n. The reason why 5.16 is the downsampling factor for a set of aligned macro-blocks is because as mentioned earlier, if there are frequency components in $\frac{\pi}{D} < \omega \leq \pi$, then aliasing occurs. By Definition 32, there are frequency components when $\omega < \omega_c$, therefore if $\frac{\pi}{D} < \omega_c$ (i.e. $D > \frac{\pi}{\omega_c}$), then there will be frequency components in $\frac{\pi}{D} < \omega \leq \pi$, hence aliasing occurs.

A set of downsampling factors $\{D_i\}$ for each macro-block is now available. The single downsampling factor that will ensure aliasing of all sets of macro-blocks is the maximum downsampling factor as given by Equation 5.17.

$$D = max_i\{D_i\} \tag{5.17}$$

The semantic class is formed by frame-wise downsampling of the original video by a factor D, given by Equation 5.17, which results in an aliased video.

2. Obtaining the Fingerprints

Once the semantic class is created via downsampling with the downsampling factor found in the previous section, the feature class is automatically those frames that are not included in the semantic class. Now the task is to assign frames from the feature class to different users, so that any user who has frames from the semantic class as well as frames from the feature class, can construct a video that looks visually similar to the original video. The question is how to partition the feature class to achieve this goal? The answer lies in the use of motion vectors found in the field of video compression research.

In video compression, motion is encoded using what are known as *motion vectors.* The MPEG 1/2 standard uses the translation model for encoding motion [46]. A video is sequence of frames, such that consecutive frames usually give rise to moving objects, which can be modeled as translation motion.[4] The motion vector gives the magnitude and direction of the translation motion of a macro-block between two frames. Figure 16 illustrates the concept of a motion vector. In Figure 16, the blue macro-block experiences a translation motion between the two consecutive frames. In Frame $i+1$,

[4] This is a good approximation to motion, although other models that incorporate zooming motion, rotational motion, etc. do exist.

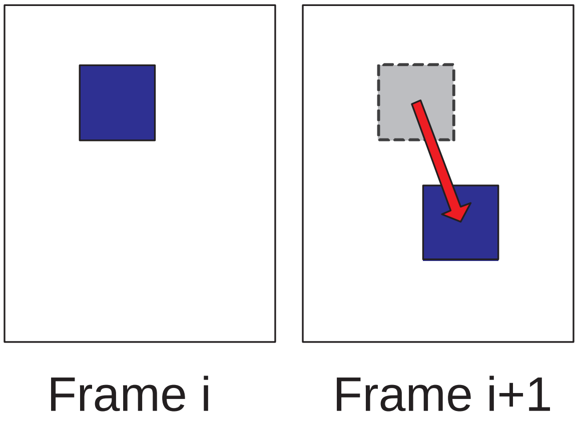


Fig. 16. Illustration of a Motion Vector Between Two Consecutive Frames

the original position of the macro-block is faded and dotted, while the new position of the macro-block is blue. The red motion vector describes this translation motion. Techniques for finding motion vectors are found in the video compression literature, such as [46]. In this work, the 2-*D logarithmic search* is implemented.

The average motion between two frames gives a good indication of whether the two frames are redundant (that is, there is not much motion occurring between the two frames), too far apart (that is, there is too much translation motion between the two frames, so playing the two frames consecutively will result in choppy motion), or just enough motion (that is, there is little redundancy, and it is visually acceptable to the human eye). The average motion vector between two frames is now derived.

Suppose that with each macro-block B_i in the first frame, there is a motion vector $\vec{m}_i$ that describes the translation motion of B_i from the first frame to the second frame. The average motion, AM between two frames is then the average magnitude of all motion vectors, given in Equation 5.18.

$$AM = \sum_i \|\vec{m}_i\| \tag{5.18}$$

Whether two frames are redundant, too far apart, or just enough, can be determined by comparing the average motion of the two frames to some threshold. For example, given two frames, if their $AM > \tau_{am}$, then the two frames are too far apart, while $AM < \epsilon$ means the two frames are redundant. Using this idea, the frames from the feature set can be chosen so that AM between consecutive frames is maintained around some constant. The fingerprinting algorithm is now described formally.

Let Ξ be the semantic class, or set of frames derived as in the previous section. Then Φ is the feature class consisting of frames not in Ξ. From Φ, frames are chosen to maximize the AM between consecutive frames with the constraint that each AM is smaller than τ_{am}. This subset of frames will ensure that there are no redundancies, and at the same time the motion is smooth. In creating different fingerprinted videos, frames from the feature class that have been used are not reused again, therefore the fingerprints are a partition of the feature class. This is now expressed mathematically. To create fingerprint Φ_i, which is a sequence of frames (ϕ_k), choose the sequence (ϕ_k) such that

$$\begin{aligned} \forall \phi_k \in \Phi_i \subseteq \Phi^{(i)} \triangleq \Phi \setminus \left(\bigcup_{j<i} \Phi_j \right) \\ AM(\phi_k, \phi_{k+1}) = \max_{\phi_m \in \Phi^{(i)}} AM(\phi_m, \phi_{m+1}) \\ \text{with constraint } AM(\phi_k, \phi_{k+1}) \leq \tau_{am} \end{aligned} \tag{5.19}$$

The first line in Equation 5.19 ensures that the frames in Φ_i have not been used by previous fingerprints; in addition, $\Phi^{(i)}$ defines the part of the feature class that has not been used by other fingerprints. The second and third lines in Equation 5.19 ensure that the average motion between frames is as far apart as possible without being too far apart, dictated by the threshold constraint.

All fingerprinted videos should have the same number of frames as the original video, to ensure that audio accompanying the video sequence is synchronized with the the video. This is achieved by simply repeating frames whenever there are missing frames. Mathematically, recall that each fingerprint has an associated binary codeword. A 0 in the codeword means the frame in that position is not included in the fingerprint Φ_i. This frame position should be filled in by the frame associated with the first 1 to the left of any 0. For example, let $(\phi_1, \phi_2, \phi_3, \phi_4)$ be a sequence of frames, and suppose that the binary codeword γ^i associated fingerprint Φ_i is $\gamma^i = 1010$. Then the following frames are played in order: $\Phi_i = (\phi_1, \phi_1, \phi_3, \phi_3)$. If there is no 1 to the left of a 0, the closest frame from the semantic class is used. For example, if $\gamma^i = 0101$, then suppose $\xi \in \Xi$ is the closest frame to ϕ_1. Then the following frames are played in order: $\Phi_i = (\xi, \phi_2, \phi_2, \phi_4)$.

3. Detection of Fingerprints

The proposed fingerprint detection scheme will only return one culprit from the coalition. An attacked video sequence is compared with all pristine fingerprinted copies, and the pristine fingerprinted copy "closest" to the attacked video is selected as being part of the coalition. To reduce the computational load, only the fingerprints Φ_i are compared, while the semantic class is not compared, since the semantic class is the same in all copies. The semantic class can however be used to estimate the attack noise, since all individuals have a copy of the semantic class, and every frame should be attacked the same way in order not to introduce visual distortion. This is similar to using a reference watermark to estimate noise parameters in [25]. Attack noise estimation using the semantic class is beyond the scope of this thesis.

The detector will be a "minimum distance detector", such that the pristine fingerprint Φ_i is chosen whenever it has the "minimum distance" with the attacked fingerprint $\tilde{\Phi}$. Here, distance is not measured via the Euclidean distance, but rather the average motion AM distance. For example, if a Euclidean-type distance is used as the minimum distance receiver, then the decision would be made according to Equation 5.20.

$$\text{user } i \text{ is a suspect} = \arg\min_{i} \sum_{f\in\Phi_i, g\in\tilde{\Phi}} \|f-g\|^2 \tag{5.20}$$

Empirical evidence shows that using the Euclidean distance for the proposed algorithm will not suffice, because all the frames are highly correlated. For example, the correlation coefficient between frame 1 and frames 1 to 35 for the test video used in the next section is provided in Figure 17. It can be seen that frame 1 is highly correlated

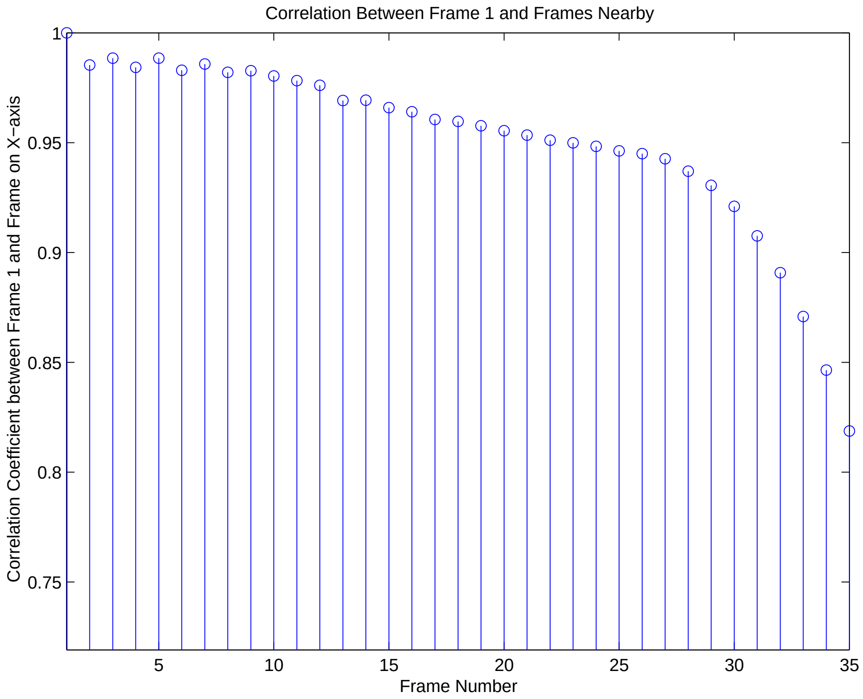


Fig. 17. Correlation Between Frame 1 and Frames Nearby

with its neighbours. The minimum distance detector using the Euclidean distance is actually the same as a maximum correlation receiver when the noise added to a signal is 0-mean, white, and Gaussian. Therefore since all frames are highly correlated, a correlator cannot adequately distinguish one fingerprinted video from another.

The key to finding the right distance metric is in recalling that each fingerprinted video differs from the others in that there is some small motion imperceptible to the human eye. However, detection of this motion is possible using the average motion as a metric for distance. Therefore the detector should be given according to Equation 5.21.

$$\text{user } i \text{ is a suspect} = \arg\min_i \sum_{f\in\Phi_i, g\in\tilde{\Phi}} AM(f,g) \tag{5.21}$$

Further justification of 5.21 can be found in Appendix D.

Section E will show that this means of detection is viable against selected single-user attacks and selected collusion attacks. In the next section, a method for supporting a larger set of users is presented, because the fingerprinting capacity of the proposed algorithm is not large.

4. Supporting a Larger Set of Users

The reader will note that the number of users that can be supported by the proposed algorithm is not large. One alternative method to increasing the number of unique copies is to combine other fingerprinting schemes with the proposed algorithm. For example, future work might focus on the integration of the JSF scheme with the watermarking scheme, such that watermarks are embedded in the semantic class, or other locations of a JSF fingerprinted video. This section will propose a method that does not incorporate other fingerprinting schemes.

The following algorithm is inspired by the outer code used in [3]. The idea is that given only n fingerprinted copies, $N \gg n$ copies can be created by partitioning each copy into L disjoint sets of frames; these frames can be adjacent frames or non-contiguous frames. A new fingerprinted video is then created for each of the N users by assembling L partitions, where each partition is randomly chosen from one of the original 1 to n fingerprinted videos corresponding to that partition. This can be thought of as using the original n fingerprinted copies as the alphabet in creating a new n-ary codebook W, whose codewords $w_i \in W$ are of length L, and represent the fingerprint for User i. Although it is possible to create n^L new fingerprinted copies from the original n copies, only a smaller subset, namely N, of these copies is used, because distributing all n^L copies will result in no collusion-resistant properties.

Detection will involve detecting the n original fingerprints (using the original detector) in each of the L partitions, and from this, forming a n-ary word of length L, denoted $\tilde{w}$. The codeword $w_i \in W$ that is closest to $\tilde{w}$ in terms of the maximum number of positions that match, is selected as a suspect. Assuming that the original n fingerprinted copies have negligible probability of error when collusion attacks are applied, the new probability of error is bounded by Equation 5.22, where c is the maximum coalition size, and is chosen such that $c < \frac{n}{2}$.

$$\epsilon \leq N \cdot Q\left(\sqrt{\frac{L}{n-1}}\left(\frac{n}{c}-1\right)\right) \tag{5.22}$$

Here $Q(\cdot)$ is the tail probability of a standard Normal distribution (zero-mean, unit-variance Gaussian) as given by Equation 5.23.

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)\, dx \tag{5.23}$$

The proof for the bound in Equation 5.22 can be found in Appendix D.

5. Efficient Broadcasting

It may seem that a naive many-to-many broadcasting scheme is the only possible solution, given all fingerprinted copies are different; that is, a unique fingerprinted video is sent to every subscribed user. However, inherent in the JSF paradigm, is the partitioning of the source data into a public class of data (semantic class), which all users receive, and a private class of data (feature class), which constitutes the unique fingerprints. Therefore the semantic class only needs to be broadcast once to all subscribed users. All the fingerprints can be broadcast to all users, with each fingerprint encrypted so that only the destined user can decrypt her fingerprint using her corresponding private key. This method of broadcasting the fingerprints is bandwidth inefficient, since all users receive all fingerprint frames, but can only decrypt a subset of these frames, hence the rest of the frames are wasting bandwidth. The solution is to use motion vectors again.

In video compression, one of the principle agents responsible for high compression rates is the encoding of frames using motion vectors [46]. This technique, referred to as motion-compensated predictive coding, is described in detail in [46], but will be succinctly summarized here. As mentioned earlier, the motion of a block B_i between two frames, can be described using motion vectors $\vec{m}_i$. In addition, an error matrix $\underline{\epsilon}_i$ is a matrix with the same dimensions as the block B_i, containing the difference between the two blocks associated by a motion vector. Hence for each block in the first frame, the corresponding translated block in the second frame can be reconstructed using the motion vector and error matrix. Since the values in the error vectors are typically much smaller than the values in the original block, compression is achieved. Therefore when frames from the feature class are broadcast to users, only encrypted pairs of motion vectors and error matrices are sent, thus reducing bandwidth. The broadcasting scheme is summarized by the equations below.

$$
\begin{align}
\text{Compression} \quad &: \quad Compress(\Phi_i) \tag{5.24}\\
&= \quad Compress(\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}) \tag{5.25}\\
&= \quad (\{(\vec{m}_j, \underline{\epsilon}_j)\}_{(i,1)}, \{(\vec{m}_j, \underline{\epsilon}_j)\}_{(i,2)}, \ldots, \{(\vec{m}_j, \underline{\epsilon}_j)\}_{(i,n)}) \tag{5.26}\\
\text{Broadcast} \quad &: \quad E_{K_{everyone}}(\Xi) \tag{5.27}\\
& \quad\;\; \{E_{K_i}(Compress(\Phi_i))\} \tag{5.28}\\
\text{Keys for User } i \quad &: \quad K_{everyone} - \text{all other users have this key} \tag{5.29}\\
& \quad\;\; K_i - \text{only User } i \text{ has this key} \tag{5.30}\\
\text{Decoding for User } i \quad &: \quad \Xi = D_{K_{everyone}}(E_{K_{everyone}}(\Xi)) \tag{5.31}\\
& \quad\;\; \Phi_i = Uncompress(D_{K_i}(E_{K_i}(Compress(\Phi_i)))) \tag{5.32}
\end{align}
$$

Symmetric-key cryptography is assumed above, that is the encryption key is the same as the decryption key, although the scheme can easily be extended to an asymmetric-key algorithm, in which the encryption key and decryption key are different. $K_{everyone}$ is a key used to decrypt the semantic class, and it is available to all subscribed users. K_i is a key available only to User i, and is used to decrypt fingerprint i.

a. Comparison of Proposed Broadcasting Scheme to Other Broadcasting Schemes

In evaluating the performance of the proposed broadcasting scheme, the performance metric used is the ratio of how much data is sent using an efficient broadcasting scheme, to how much data is sent using a many-to-many approach. For example, one possible many-to-many approach is to send N uniquely encrypted fingerprinted copies to *all* N users, so that each user will decode her own copy, while the other $N - 1$ encrypted copies are essentially "wasted". In the proposed broadcasting scheme, the only "wasted" data are the fingerprint frames Φ_i, which are only useful for User i, and not any User $j \neq i$. The performance ratio can be defined as follows. Let S be the size in bytes of one fingerprinted copy, and assume that all fingerprinted copies are of the same size. Therefore, given N users, the many-to-many approach would result in sending $N \times S$ bytes. For the proposed broadcasting scheme, assume that for each fingerprinted copy, there are F unique fingerprint frames in a total of T frames for the entire video. Again, it is assumed that all users have F unique frames, and a total of T frames. Then the semantic class would be $S - \frac{F}{T} \times S$ bytes long, which is sent to all users. Let the the fingerprinted copies be c-collusion-resistant, and extended to N users via the technique in the previous section. Therefore only $c \times \frac{F}{T}$ bytes of unique fingerprint frames are sent to all users. The performance ratio for the proposed broadcasting scheme is therefore given by Equation 5.33.

$$\begin{aligned}\text{Performance Ratio for Proposed Broadcasting} &= \frac{\left(S - \frac{F}{T}S\right) + c \times \frac{F}{T}S}{N \times S} \\ &= \frac{1 + (c-1)\frac{F}{T}}{N} \qquad (5.33)\end{aligned}$$

The proposed broadcasting scheme is compared to the broadcasting scheme in [3] (reviewed in Chapter III). For the broadcasting scheme in [3], the performance ratio is $\frac{2\times S}{N\times S} = \frac{2}{N}$. The smaller the ratio, the more efficient the broadcasting scheme, therefore the proposed broadcasting scheme is more efficient than that in [3] when Equation 5.34 is satisfied.

$$1 + (c-1)\frac{F}{T} < 2 \qquad (5.34)$$

Figure 18 shows the region for which the proposed broadcasting scheme is more efficient than the scheme in [3]. A coalition size of 20 only allows 13 unique frames out of 260 frames in order to satisfy Equation 5.34. It turns out that this is sufficient, as will be shown in Section E. In fact, the simulation results yield $c = 5$, $F = 13$, and so these quantities are well within the region in Figure 18.

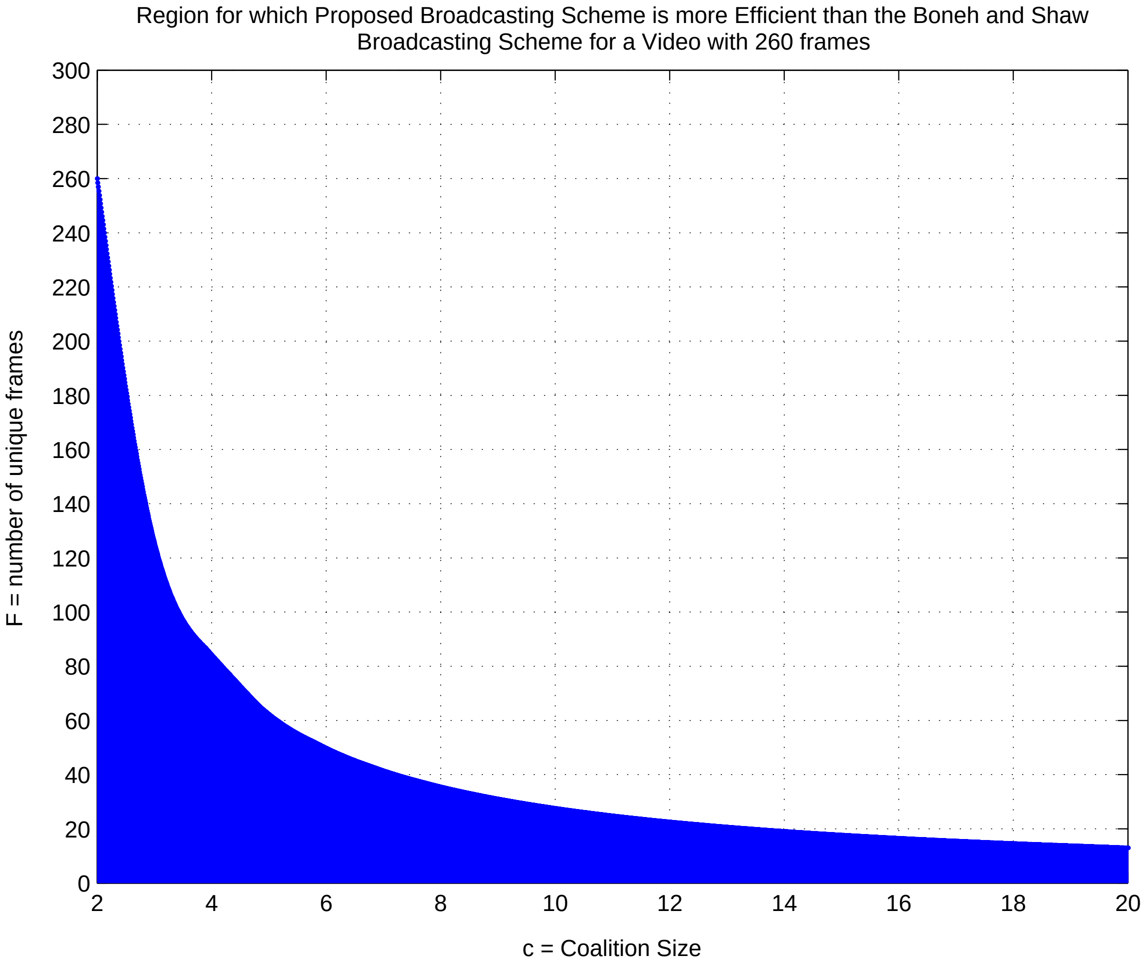


Fig. 18. Region Such That the Proposed Broadcasting Scheme Is More Efficient Than the Broadcasting Scheme in [3]

The proposed broadcasting scheme is also compared to the fingerprinting multicast scheme in [11]. However, the performance ratio in [11] is not the same as that used in this thesis, because multicast metrics are used. In [11], the performance ratio is given in Equation 5.35.

$$\gamma^{fm} = \frac{C_{multi}^{unit} \times Len_{multi}^{fm} + N \times C_{uni}^{unit} \times Len_{uni}^{fm}}{N \times C_{uni}^{unit} \times Len^{pu}} \tag{5.35}$$

The C_{multi}^{unit} and C_{uni}^{unit} are costs associated with using the multicast and unicast channels. The ratio $\frac{C_{multi}^{unit}}{C_{uni}^{unit}} \triangleq N^{EoS}$ with $EoS = 0.7$ is used in [11]. Len_{multi}^{fm} is associated with the size of the data that is the same for all users (i.e. the semantic class in the proposed scheme), and Len_{uni}^{fm} is associated with the size of the data that is different for all users (i.e. the unique fingerprints in the proposed scheme). Finally Len^{pu} is associated with the size of one fingerprint copy if no clever broadcast scheme is used. Therefore, if the cost metrics C_{multi}^{unit} and C_{uni}^{unit} are are set to 1, then the performance ratio in Equation 5.35 would be the same as that used in this thesis. Unfortunately the results in [11] are obtained empirically through simulation, so a closed-form expression is not available. However, some bounds can be established such that the results in [11] can be compared with the performance of the proposed scheme in this thesis. The right-hand side in Equation 5.36 is the performance ratio used in this thesis. It can be shown that multiplying the ratio in Equation 5.35 by $N^{-0.7}$ will result in a performance ratio less than that used in this thesis. Recall that a smaller ratio means better efficiency, therefore if the proposed scheme has a smaller ratio than those in [11] multiplied by the factor $N^{-0.7}$, then the proposed scheme is more efficient.

$$N^{-0.7}\gamma^{fm} < \frac{Len_{multi}^{fm}}{N \times Len^{pu}} + \frac{Len_{uni}^{fm}}{Len^{pu}} \tag{5.36}$$

Figure 19 compares the efficiency of the proposed broadcasting scheme to that in

[11]. The most efficient curve from [11] (Miss America video) is used. Recall that this curve is the left-hand side of Equation 5.36, which means the true data is less efficient than depicted in Figure 19. For $(c, F) = (20, 13)$, the proposed scheme might be less efficient than that in [11]. However for $(c, F) = (5, 13)$, the proposed scheme is more efficient than that in [11]. It should be noted however, that the proposed scheme is c-collusion-resistant for small c, while the scheme in [11] is c-collusion-resistant for larger c, but [11] lacks JSF properties, such as punishing colluders with visual

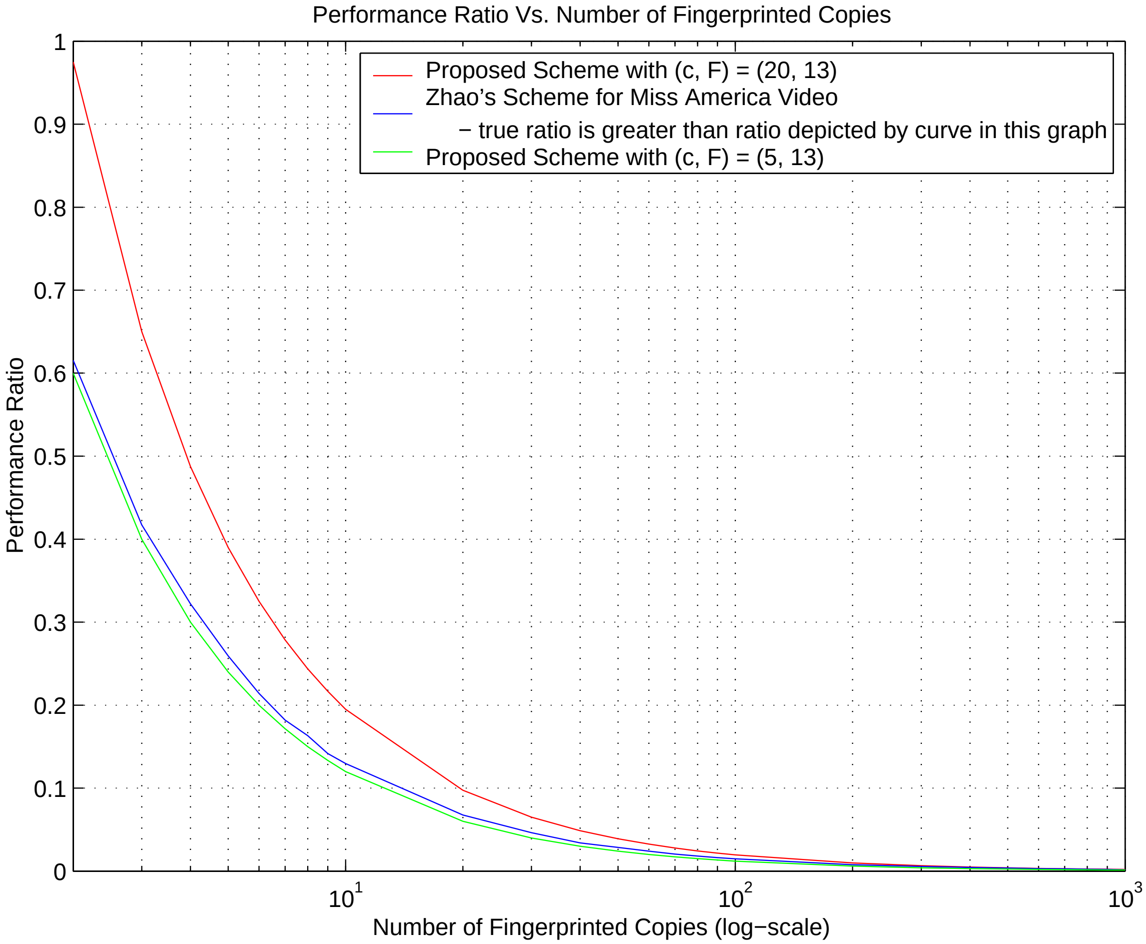


Fig. 19. Comparison of Broadcast Efficiency Between Proposed Scheme and Scheme Found in [11]

degradation.

E. Simulation Results

In testing the fingerprinting algorithm and detection algorithm from Section D, 12 out of 20 videos from the Video Quality Experts Group (VQEG) were selected. These 12 videos capture motion ranging from translation, zooming in and out, to little motion, slow motion and fast motion.[5] In this section, the test video used is that of a woman talking on the phone. The reason why this video is used is because there is very little motion, so it is actually easier to attack; this makes the results from this input video, somewhat of a lower bound on performance. Recall that the detector utilizes average motion, so a video with little motion will result in very small average motions, which is a poor condition for the detector. The results will show that even though this input is the weakest of the 12 videos, correct identification of one pirate, as well as punishment via visual degradation is achieved. Additional simulation results can be found in Appendix E.

The tests performed are categorized into single-user attacks, and collusion attacks. The single-user attacks include JPEG compression, AWGN, translation of random macro-blocks, and a rotation attack. The collusion attacks consist of averaging, random scrambling, and the order statistics attacks given by Equations 2.11 to 2.16. Comparison of results with other algorithms is limited to showing how the JSF algorithm punishes colluders with visual degradation, while the traditional watermarking algorithms do not, since this is one of the main contributions of the JSF paradigm.

[5] The other 8 videos were not fit for testing because they were animations, or computer rendered graphics.

1. Robustness to Single-User Attacks

In the single user attack, the goal of the attacker is to perturb a frame, so the perturbed frame is far apart from the original frame in terms of average motion. The results show that the specified attacks cannot feign motion.

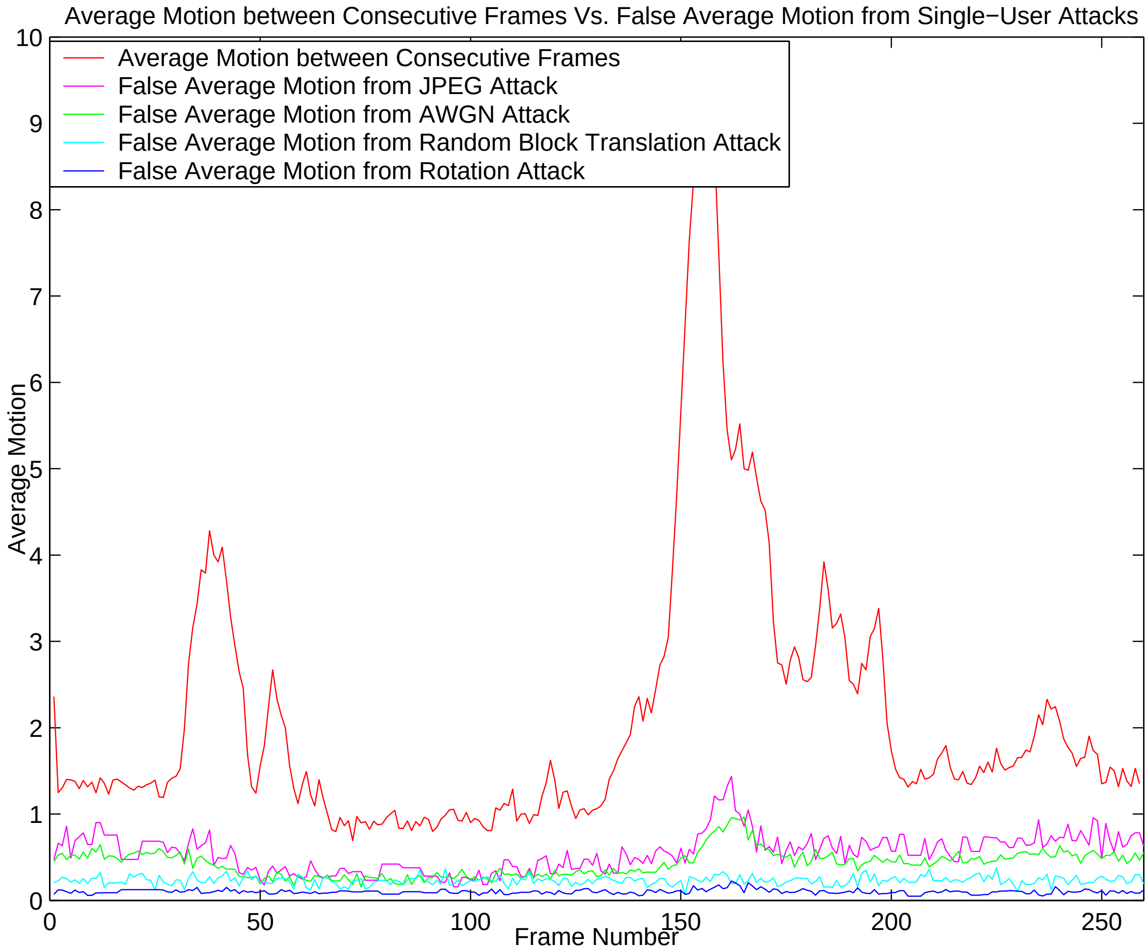


Fig. 20. Average Motion Between Consecutive Frames and False Average Motion From Single-User Attacks

In Figure 20, the top-most curve (red) depicts the average motion between consecutive frames in the test video. The curves below show the average motion between the original frame, and the attacked frame. It can be seen that these attacks do not simulate motion, as their average motions are very small, and usually less than 1.

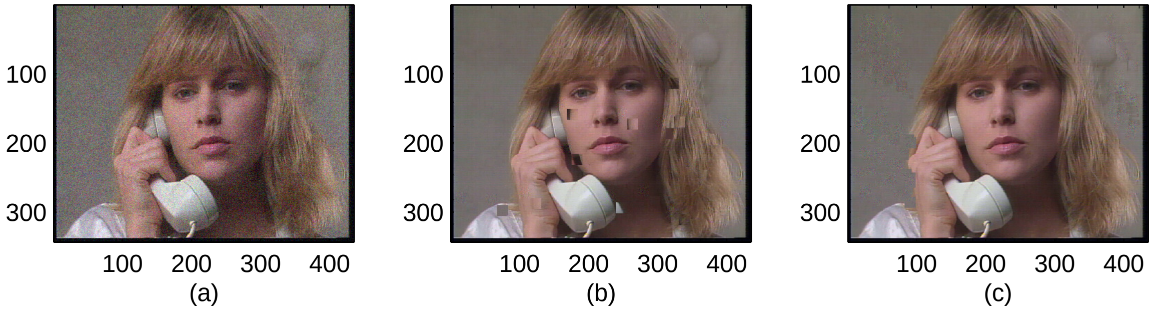


Fig. 21. (a) AWGN Attack with Variance 0.005 on Each RGB Colour Plane; (b) Random Block Translation Without Restrictions; (c) Random Block Translation Restricted to Background

The specifics of the attacks are now described. In the JPEG attack, each frame is compressed using the JPEG write option from MATLAB. The quality is set at 40%, which will give a little visual distortion. Settings of lower quality, such as a value of 30% result in a lot of visible blockiness. From Figure 20, the JPEG compression attack is the most effective out of the other attacks tested, since its average motion curve is higher than the other curves. To mitigate this attack, frames can be compressed at a visually acceptable level prior to fingerprinting, therefore the attacker cannot compress the frames more without introducing visual distortion.

The additive white Gaussian noise (AWGN) attack was described in Chapter II. The variance used in the simulation is 0.001 applied on each of the RGB colour planes. Any other variance used, produces visible distortions. For example, Figure 21(a) shows the first frame being attacked with a variance of 0.005. Even the tested variance of 0.001, produces visible distortion when the frames are played consecutively. Again, the AWGN attack does not mimic motion, and the attacked frames are close to the original frames in the average motion sense.

The random block translation attack is geared especially at crippling the minimum average motion distance detector. It is hoped that by translating random blocks, the resulting frame will be far away from the original frame in terms of average motion, since the translation model is the underlying assumption in computing the average motion. However, the problem with this attack is that most of the time moving a macro-block will easily affect the visual quality of the frame. Therefore the attacker must spend more computational energy in finding parts of the frame that are immune to visual degradation when blocks are moved, such as any uniform background. Even when such uniform backgrounds are identified, which in the first place may not exist, the number of blocks that can be translated without visual degradation is low. Figure 21(b) shows a random block translation without restrictions, hence visually degrading the face, while 21(c) shows the background becoming visually noisy when 100 blocks are translated. This means only a few blocks can be translated without visual distortion. In simulating such an attack without visual distortion, 10 blocks were randomly translated. The results in Figure 20 show that this does not mimic true motion, since the average motion is close to 0.

The final single-user attack tested is the rotation attack. There are again restrictions which the attacker must abide to, or the consequence is a visually distorted video. For example, all frames must be rotated the same way, or very close, otherwise when played, the video will appear choppy. In addition, the rotation cannot be too large, thus the attacker will make a rotation of no more than 1 degree. The first restriction that all frames have to be rotated the same way, allows the receiver to estimate the rotation incurred. For example, all fingerprinted copies share the same semantic class frames, so these frames can be used to estimate the rotation of all frames. Once this estimate is available, the receiver can undo the rotation, by applying a reverse rotation. The test performs rotation estimation and reverse rotation before the decision is made. Figure 20 shows that the small errors incurred in reversing the rotation are not enough to mimic true motion, and hence these attacked frames are close to the original frames in the average motion sense.

2. Robustness to Collusion

Continuing with the video used in the previous section, 5 fingerprinted videos are generated using the proposed algorithm. All 5 fingerprinted videos appear visually identical when played, and each copy has 13 unique frames, which when removed, results in a video with visual degradation - i.e. the motion is choppy and not smooth.

All $2^5 - \binom{5}{0} - \binom{5}{1} = 26$ coalitions of two or more fingerprinted videos are tested against collusion attacks. The proposed minimum distance detector is then used to determine one of the suspects.

The results for all 26 average attacks are presented in Figure 22. For example, in the top left corner, the graph shows 5 bars. Two bars, 4 and 5, are red, meaning fingerprinted copies 4 and 5 were averaged, resulting in the collusion attacked video; the blue bars correspond to innocent users who do not participate in the average attack. The bars themselves show the AM distance between the attacked video and the pristine copy. For example, bar 1 in the same top left corner graph is approximately 20 - this means the AM distance between copy 1 and the attacked video is approximately 20. Recall that the minimum distance receiver will choose the bar with the lowest value. In the top left corner graph, the lowest value is bar 4. This also happens to be one of the colluders, since its bar is red, therefore the receiver successfully identifies one suspect. Looking at all the other graphs, the minimum distance receiver always successfully identifies one culprit, as the bar with the smallest value is always red. In addition, the red bars are always smaller than the blue bars.

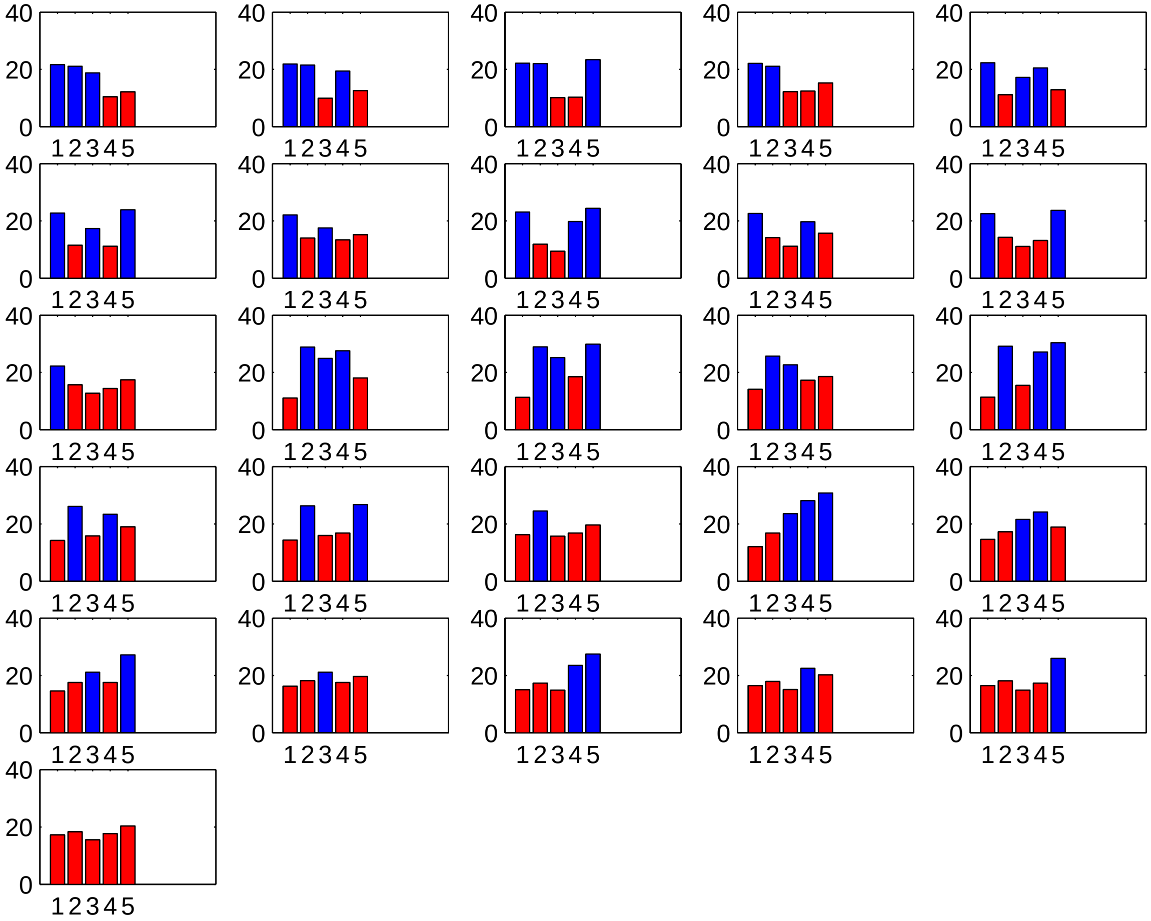


Fig. 22. Y-axis on Each Graph Is the Average Motion Distance Between the Average Attacked Video (Averaging of Fingerprinted Videos Whose Bars Are Red) and the Fingerprinted Video (Whose Number Is on the X-axis)

In addition to studying the ability of the detector to correctly identify one pirate, these simulation results also show how the proposed algorithm causes visual degradation as punishment for collusion. In Figure 23, a frame appears blurry after an average attack, hence reducing its commercial value. In the traditional paradigm, where a fingerprint codeword is embedded into the frame using a DCT-based watermarking technique [1], the average attacked frame appears identical to the original frame.

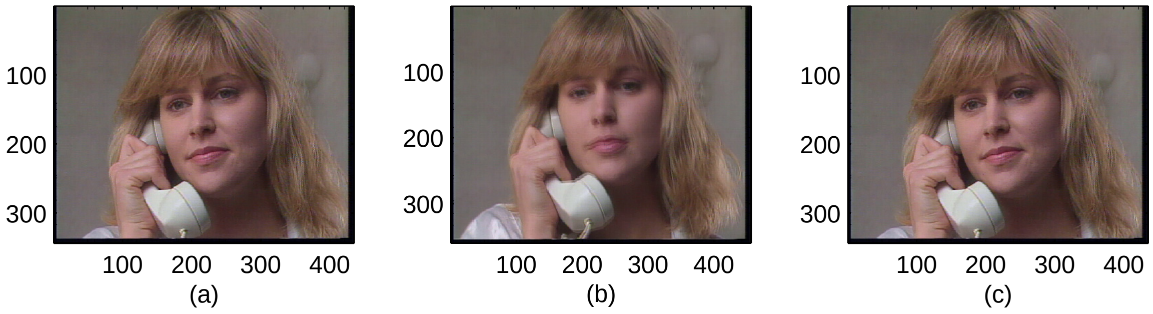


Fig. 23. (a) Original Frame; (b) Blurry Frame After Average Attack; (c) Average Attack on 60 Watermarked Frames

All 26 coalitions of two or more fingerprinted videos are tested against the random scrambling attack given by Equation 2.9. Recall that in this attack, a pixel is randomly chosen from the coalition's collection of pixels corresponding to the same coordinates. The results for all 26 attacks are presented in Figure 24. Again, all suspects are identified correctly, since all the red bars have minimum average motion distance. In addition, the red bars are always smaller than the blue bars.

The proposed algorithm also causes visual degradation as punishment for collusion. In Figure 25, the eyes become distorted after the random scrambling attack. In the traditional paradigm, where a fingerprint codeword is embedded into the frame using a wavelet-based watermarking technique [22], the random scrambling attacked

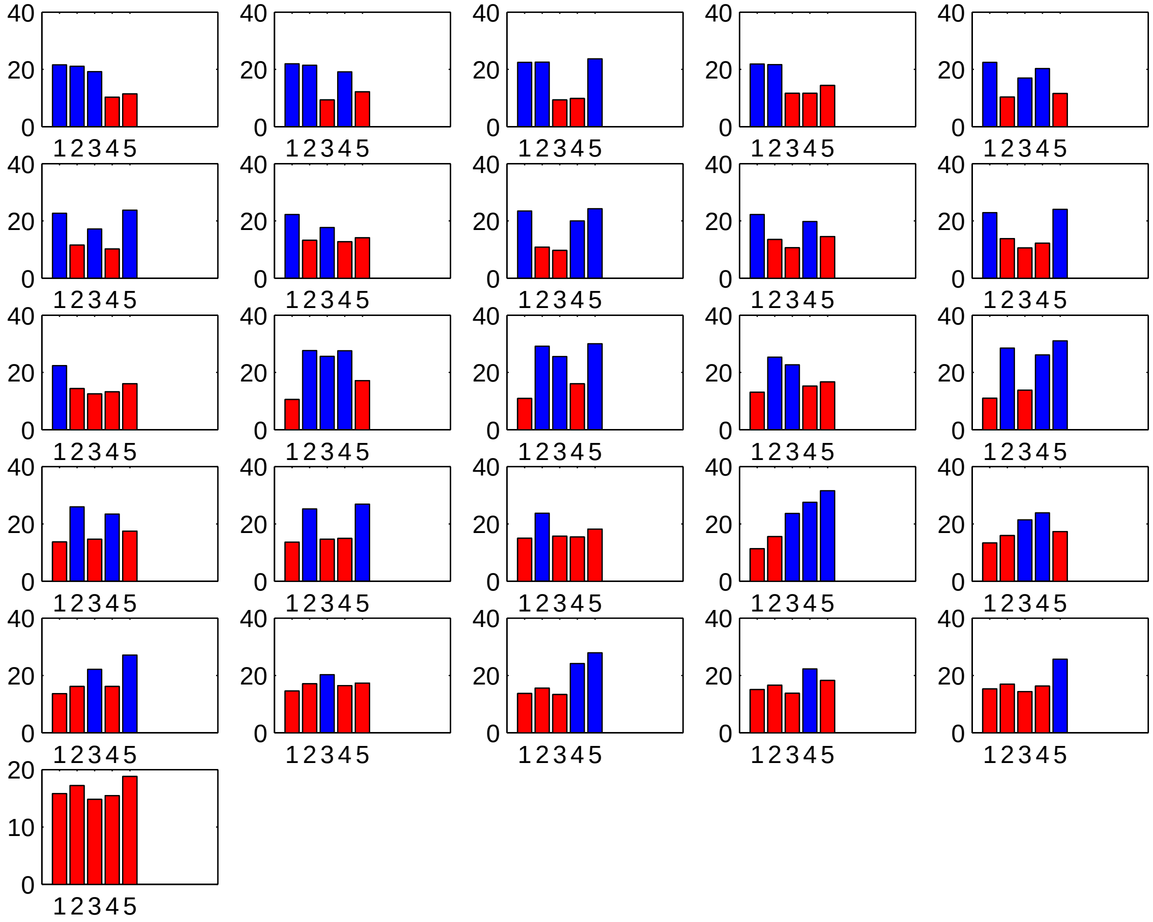


Fig. 24. Y-axis on Each Graph Is the Average Motion Distance Between the Random Scrambling Attacked Video (Random Scrambling of Fingerprinted Videos Whose Bars Are Red) and the Fingerprinted Video (Whose Number Is on the X-axis)

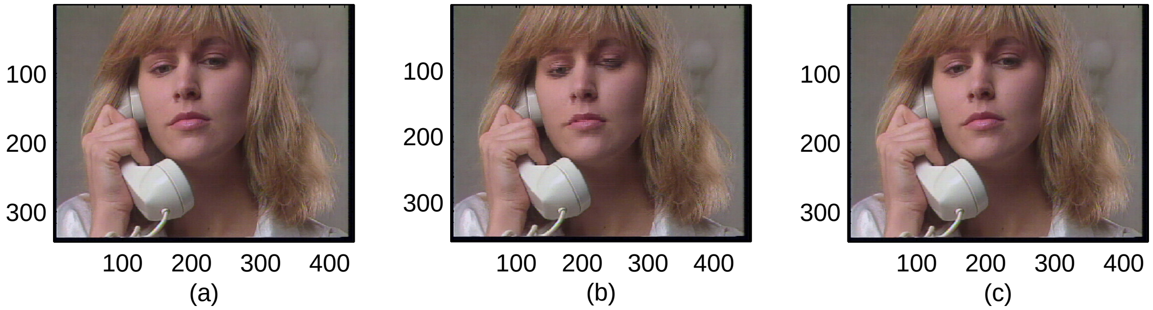


Fig. 25. (a) Original Frame; (b) Distortion Over Eyes After a Random Scrambling Attack; (c) Random Scrambling on 60 Watermarked Frames

frame appears identical to the original frame. This is because all frames are perfectly synchronized in the sense that any salient features, such as the eyes, are at the same position across all fingerprinted copies. In the proposed algorithm, the lack of synchronization brings about the property of visual degradation as punishment for collusion.

All the nonlinear attacks described by Equations 2.11 to 2.16 are also implemented. The results are all the same in that one suspect is always identified correctly, and the culprits always have the smallest average motion distance. In particular, for comparison purposes, the results for the randomized negative attack is presented in Figure 26. Recall that this attack randomly chooses either the maximum pixel value or minimum pixel value. In [21], it is shown that this attack is effective at destroying the Gaussian i.i.d. fingerprints. However, here the attack has no effect, and a suspect is always correctly identified. The other nonlinear attacks are also ineffective at destroying the fingerprints created by the proposed algorithm.

Finally, further evidence in Figure 27 shows that visual degradation increases as more copies are used in the collusion attacks for the proposed algorithm. The 60 fingerprinted frames are generated by interpolating between two frames, using upsam-

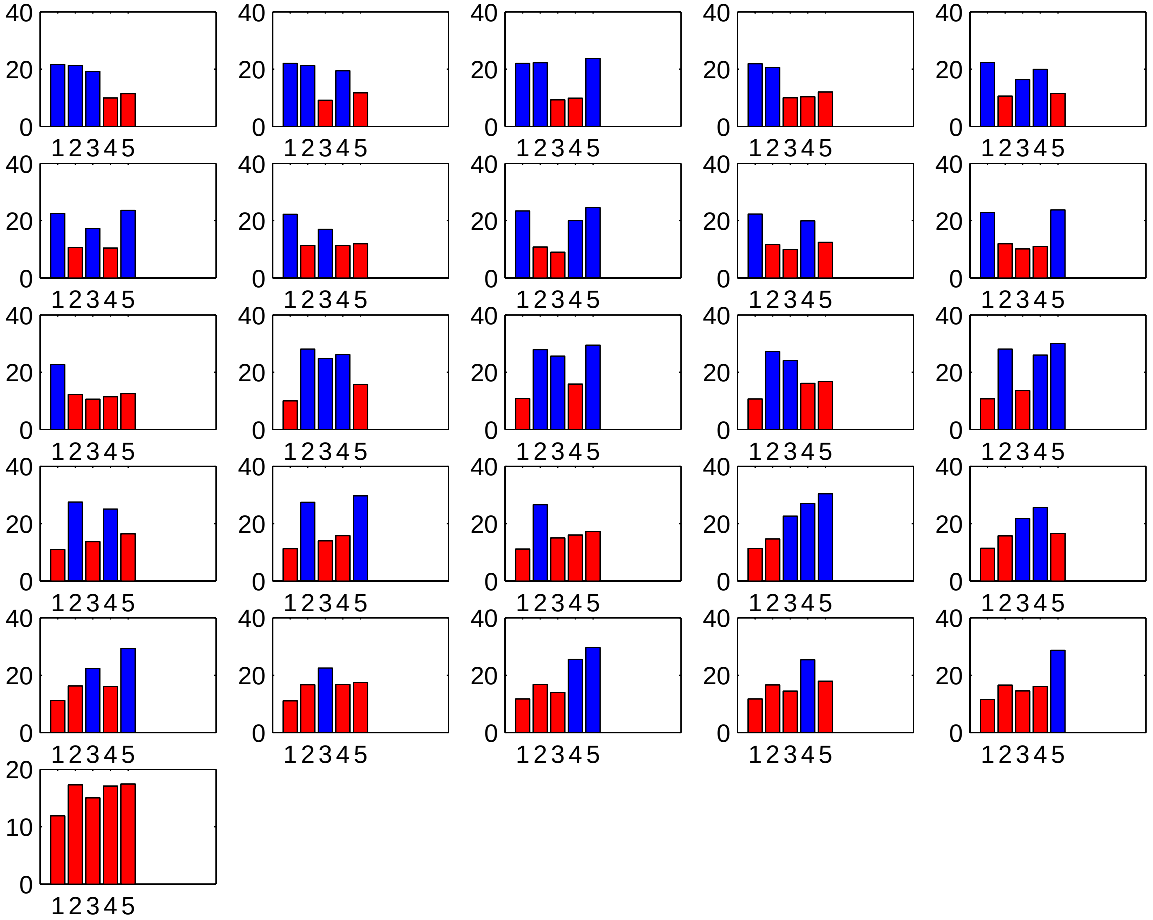


Fig. 26. Y-axis on Each Graph Is the Average Motion Distance Between the Randomized Negative Attacked Video (Randomized Negative Attack of Fingerprinted Videos Whose Bars Are Red) and the Fingerprinted Video (Whose Number Is on the X-axis)

pling and filtering, a well-known method in signal processing. Although generation of fingerprint frames via interpolation is not used in the proposed algorithm, because the resulting frames are "too close" to one another, hence leading to higher error rate, the fact that the PSNR decreases for "close frames" will imply that the PSNR will decrease even more for frames further apart. On the other hand, using watermarking

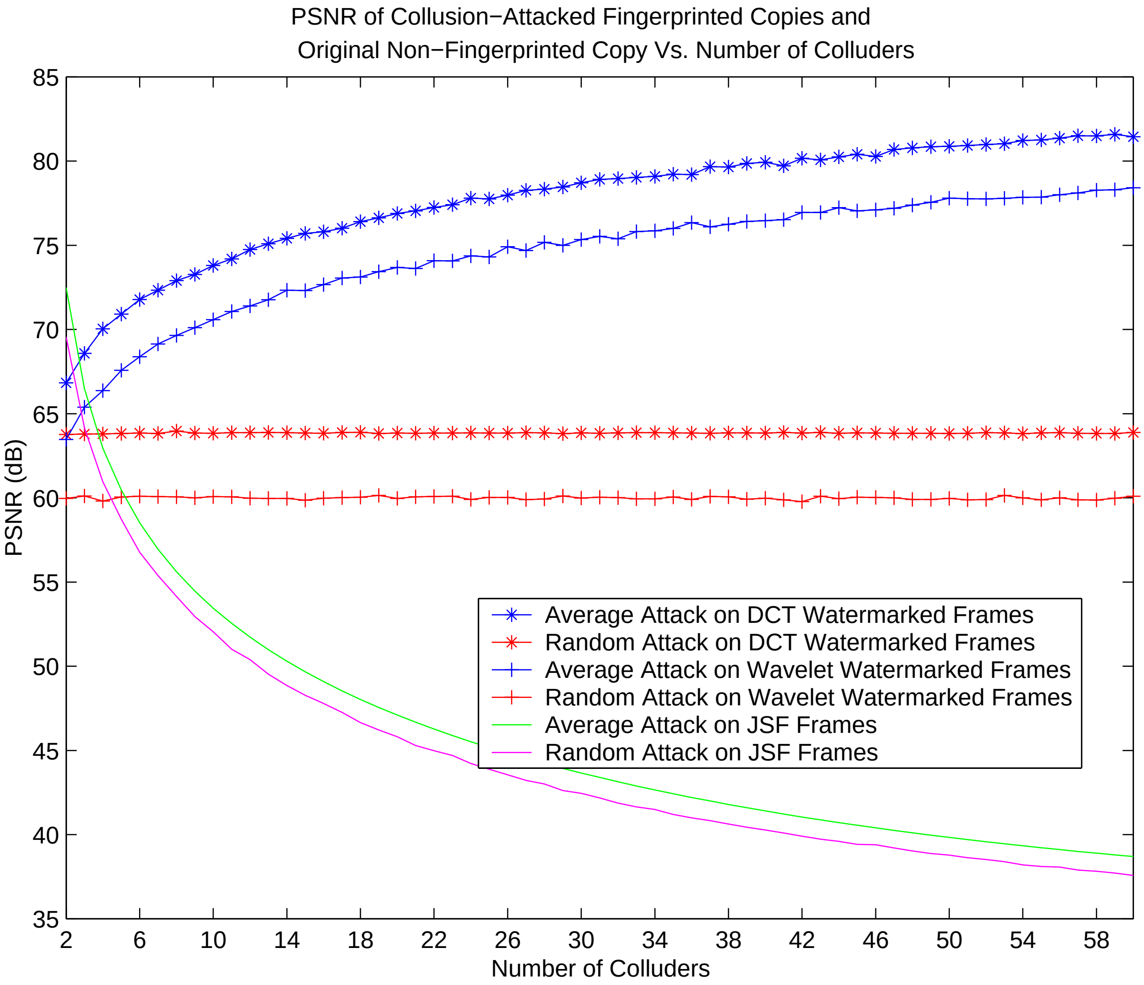


Fig. 27. PSNR vs. Number of Colluders for the JSF Video Algorithm and Watermarking Algorithms from [1] (DCT) and [22] (Wavelet).

techniques [1, 22], the attacked frame either becomes more similar to the original non-fingerprinted frames, or does not degrade in visual quality. The curves for the watermarked methods in Figure 27 are generated by running 500 random experiments for each set of colluders (2 to 60 colluders on the x-axis), and then averaging out the 500 PSNRs for each set of colluders. In the average attack, the PSNR is expected to increase, since the watermarks have zero mean, and hence averaging zero-mean i.i.d. random variables tend to zero as the number of random variables increases according to Kolmogorov's Second Strong Law of Large Numbers. For the random attack, the PSNR of the attacked watermarked frames stays constant, but does not decrease, therefore colluders are not being punished with visual degradation.

CHAPTER VI

CONCLUSIONS AND FURTHER RESEARCH

A. Conclusions

A means to combine existing codebooks such that the resulting combination is more robust to more general collusion attacks is developed. Existing codes, such as the BIBD codes [36, 4, 26], and traceability codes [28, 30, 31] can be extended to become robust to the extended narrow-sense attack, using the methods presented in this thesis. Previously, these existing codes are only robust to restrictive attacks, such as the binary AND, narrow-sense attacks without erasure, respectively. In multimedia fingerprinting, it is desirable to be robust against less restrictive attacks, since multimedia can be attacked in various ways, without affecting the visual or audio quality. The constructions presented by this thesis, hence, enhance the ability of existing codes to be used in multimedia.

In addition, compared to existing work, the proposed code excels in terms of its short codeword length when the coalition size is large. The need for large coalition sizes is justified for "malicious distributors", a problem not previously considered in the literature. Existing fingerprinting schemes assume small coalition sizes, which is appropriate if only the end-users are to collude. However, under the new scenario where there are many distributors, collusion may exist prior to distribution to the end-users. This mass collusion attack cannot be adequately traced by existing schemes, as demonstrated in this thesis.

Almost all fingerprinting schemes are based on codebook design and watermarking. While this is adequate in most cases, this thesis also presents a new paradigm, which adds an additional dimension to the criteria of a good fingerprinting scheme.

Existing watermarking schemes do not "punish" collusion attacks with visual degradation. Given enough copies are averaged for example, the resulting media is visually unaffected, while the fingerprint is most likely destroyed. The advantage offered by the novel Joint Source Fingerprinting paradigm, is that as the number of copies used in a collusion attack increases, the attacked media exhibits more visually degradation losing its commercial value. The simulation results show that the proposed algorithm for video is robust to a number of single-user attacks, as well as collusion attacks, and at the same time offers the required visual degradation from collusion attacks.

The inherent partitioning property of the semantic-feature representation for the JSF paradigm, allows fingerprinted media to be broadcast efficiently to all users, without the need to specialize transmission for individual users. It is shown that under certain conditions, the broadcasting scheme resulting from the JSF paradigm, out-performs existing broadcasting schemes that abide by the same assumptions.

A drawback of the proposed video fingerprinting algorithm, is the robustness to only a small coalition size. As explained earlier, this is suitable when collusion is considered at the end-user level. However, if malicious distributors are at play, the proposed algorithm is easily defeated.

B. Further Research

The field of digital fingerprinting is approximately 10 years old when only effective techniques are considered. Most of the research has focused on codebook design, with little interest on how the resulting codewords are embedded in the host source itself. Therefore existing watermarking techniques, most of which do not consider collusion, may not be appropriate for fingerprinting. The Joint Source Fingerprinting paradigm is novel, and hence offers various future research possibilities. Some of these possibilities are outlined here.

- The choice of the transform $T(\cdot)$ is very important in the JSF paradigm, as this determines the computational complexity of the fingerprint search problem, as well as the fingerprinting capacity. The transform leading to the most efficient fingerprinting search would be one where the visual entropies are additive. If this is true, then the fingerprinting search problem is optimal under a greedy algorithm. For example

  $$H_V(\Xi, \{\varphi_1\} \cup \{\varphi_2\} \cup \ldots \cup \{\varphi_n\}) = \sum_{i=1}^{n} = H_V(\Xi, \{\varphi_i\})$$

  which means the visual entropies of the individual $H_V(\Xi, \{\varphi_i\})$ can be calculated in advance, and the search algorithm will greedily choose the largest ones such that the resulting sum is greater than H_τ. It is also believed transform $T(\cdot)$, that leads to the condition of additivity of visual entropy, is equivalent to the the Karhunen-Loeve transform. For example, the Karhunen-Loeve transform aims to separate the input into uncorrelated components. Components that exhibit no correlation will most likely result in additivity of visual entropies.

- One possible technique that relates transform coefficients with visual quality, is the embedded zero-tree wavelet (EZW). It may be possible to use the EZW to partition wavelet coefficients into many non-overlapping fingerprints. However, using the EZW alone will result in collusion attacks that easily remove the fingerprints. Therefore a means to buffer this approach is needed.

- While finding a good transform to speed up computation is important, it is also important that the transform allows for separation of the media such that collusion attacks result in visual degradation. The Quasi-JSF paradigm can also be used, however, instead of using watermarking, a one-way function can be applied to each and every frame such that the function is able to mimic true motion, but not allow any reversal of motion, as well as any additional motion.

- The proposed video fingerprinting algorithm uses motion vectors as an equivalent to visual entropy. Newer means of motion estimation exist in the MPEG-4 and MPEG-7 standards, that allow for other motion models, instead of just the translational motion model. These methods can be applied to improve detection, since the desired detector will employ a true motion detector.

- To design a better detector, the motion PDF should be estimated, prior to distribution, and then used in the fingerprint detection stage. In this thesis, many assumptions are made as to how the PDF behaves. Although these assumptions are far less restrictive than the general normal distribution assumption made by many in this field of research, an estimation of the PDF would prove to relax these assumptions even more. At the same time, an analytical expression for the probability of error would be derivable.

- The proposed video fingerprinting algorithm can only catch one pirate by taking the lowest output from the detector. There are several ways that all pirates can be caught. The first method is to estimate the number of pirates $\hat{n}$, involved in the collusion attack, and then take the $\hat{n}$ lowest outputs from the detector. Another way to catch all users, is to use the inherent property that every fingerprinted copy has an underlying binary codeword. The codeword itself can be used to determine all colluders, much in the same way as presented in Chapter IV. This, however, would make the detector a hard-decision decoder.

REFERENCES

[1] I. J. Cox, J. Kilian, T. F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, pp. 1673–1687, December 1997.

[2] F. Deguillaume, G. Csurka, and T. Pun, "Countermeasures for unintentional and intentional video watermarking attacks," in *Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, 2000, vol. 3971.

[3] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, pp. 1897–1905, September 1998.

[4] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, pp. 15–27, March 2004.

[5] R. Safavi-Naini and Y. Wang, "Collusion secure q-ary fingerprinting for perceptual content," in *Security and Privacy in Digital Rights Management. ACM CCS-8 Workshop DRM 2001*, 2002, pp. 57–75.

[6] F. Sebe and J. Domingo-Ferrer, "Scattering codes to implement short 3-secure fingerprinting for copyright protection," *Electronics Letters*, vol. 38, no. 17, pp. 958–959, 2002.

[7] G. Tardos, "Optimal probabilistic fingerprint codes," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003, pp. 116–125.

[8] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *SPIE Journal of Electronic Imaging*, vol. 9, pp. 456–467, 2000.

[9] J. Domingo-Ferrer and J. Herrera-Joancomarti, "Simple collusion-secure fingerprinting schemes for images," in *2000 International Symposium on Information Technology*, March 2000, pp. 128–132.

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

[11] H. Zhao and K. J. R. Liu, "Bandwidth efficient fingerprint multicast for video streaming," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2004.

[12] R. Anderson and C. Manifavas, "Chameleon - a new kind of stream cipher," in *Fast Software Encryption*, Haifa, Israel, January 1997, pp. 107–113.

[13] I. Brown, C. Perkins, and J. Crowcroft, "Watercasting: Distributed watermarking of multicast media," in *Network Group Communications*, November 1999, pp. 286–300.

[14] P. Q. Judge and M. H. Ammar, "Whim: Watermarking multicast video with a hierarchy of intermediaries," in *Proc. NOSSDAC*, June 2000, pp. 699–712.

[15] H. Chu, L. Qiao, and K. Nahrstedt, "A secure multicast protocol with copyright protection," *ACM SIGCOMM Computer Communications Review*, vol. 32, no. 2, pp. 42–60, April 2002.

[16] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, 2nd edition, New York: Morgan Kaufmann Publishers, 2002.

[17] D. R. Stinson, *Cryptography: Theory and Practice*, 1st edition, Chicago: Chapman and Hall, 1995.

[18] D. Kundur and K. Karthik, "Digital fingerprinting and encryption principles for digital rights management," *Proc. of the IEEE Special Issue on Enabling Security Technologies for Digital Rights Management*, vol. 92, no. 6, pp. 918–932, June 2004.

[19] A. Barg, G. R. Blakley, and G. A. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Transactions on Information Theory*, vol. 49, pp. 852–865, April 2003.

[20] Z. J. Wang, M. Wu, W. Trappe, and K. J. R. Liu, "Group-oriented fingerprinting for multimedia forensics," *EURASIP Journal on Applied Signal Processing*, vol. 14, pp. 2153–2173, 2004.

[21] H. Zhao, M. Wu, Z. J. Wang, and K. J. R. Liu, "Nonlinear collusion attacks on independent fingerprints for multimedia," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, pp. 613–616.

[22] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 525–540, 4 1998.

[23] B. Pfitzmann and M. Schunter, "Asymmetric fingerprinting," *Lecture Notes in Computer Science*, vol. 1070, pp. 84–95, 1996.

[24] B. Pfitzmann and M. Waidner, "Anonymous fingerprinting," *Lecture Notes in Computer Science*, vol. 1233, pp. 88–102, 1997.

[25] D. Kundur and D. Hatzinakos, "Diversity and attack characterization for improved robust watermarking," *IEEE Transactions on Signal Processing*, vol. 29, no. 10, pp. 2383–2396, October 2001.

[26] W. Trappe, Min Wu, and K.J.R. Liu, “Technical research report: Anit-collusion fingerprinting for multimedia,” *Institute for Systems Research University of Maryland*, 2002.

[27] V. D. To, R. Safavi-Naini, and Y. Wang, “A 2-secure code with efficient tracing algorithm,” in *Proceedings in the Third International Conference on Cryptology*, 2002, pp. 149–162.

[28] B. Chor, A. Fiat, M. Naor, and B. Pinkas, “Tracing traitors,” *IEEE Transactions on Information Theory*, vol. 46, no. 5, pp. 893–910, 2000.

[29] D. Kirovski, H. S. Malvar, and Y. Yacobi, “Multimedia content screening using a dual watermarking and fingerprinting system,” *IEEE Multimedia Magazine*, vol. 11, no. 3, pp. 59–73, 2004.

[30] J. N. Staddon, D. R. Stinson, and R. Wei, “Combinatorial properties of frameproof and traceability codes,” *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 1042–1049, 2001.

[31] A. Silverberg, J. Staddon, and J. Walker, “Applications of list decoding to tracing traitors,” *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1312–1318, 2003.

[32] F. Ergn, J. Kilian, and R. Kumar, “A note on the limits of collusion-resistant watermarks,” *Advances in Cryptology*, vol. 17, pp. 140–149, 1999.

[33] G. R. Blakley, C. Meadows, and G. B. Purdy, “Fingerprinting long forgiving messages,” in *Lecture notes in computer sciences - On Advances in cryptology*, 1986, vol. 218, pp. 180–189.

[34] A. Beautelspacher and U. Rosenbaum, *Projective Geometry*, Cambridge: Cambridge University Press, 1998.

[35] J. W. P. Hirschfeld, *Projective Geometries over Finite Fields*, Oxford: Oxford University Press, 1998.

[36] W. Trappe, Min Wu, Z. J. Wang, and K. J. R. Liu, "Anti-collusion fingerprinting for multimedia," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1069–1087, April 2003.

[37] J. H. Dinitz and D. R. Stinson, *Contemporary Design Theory: A Collection of Surveys*, New York: John Wiley and Sons, 1992.

[38] C. C. Linder and C. A. Rodger, *Design Theory*, Boca Raton, FL: CRC Press, 1997.

[39] C. J. Colbourn and J. H. Dinitz, *The CRC Handbook of Combinatorial Designs*, Boca Raton, FL: CRC Press, 1996.

[40] R. Parviainen and P. Parnes, "Large scale distributed watermarking of multicast media through encryption," in *Proc. of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security Issues of the New Century*, 2001, vol. 64, pp. 149–158.

[41] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, New York: John Wiley and Sons, 1991.

[42] K. Su, D. Kundur, and D. Hatzinakos, "A content-dependent spatially localized video watermark for resistance to collusion and interpolation attacks," in *Proc. IEEE Int. Conf. on Image Processing*, October 2001, vol. 1, pp. 818–821.

[43] M. Barni, F. Bartolini, A. De Rosa, and A. Piva, "Capacity of the watermark channel: how many bits can be hidden within a digital image," in *Proc. SPIE*, January 1999, vol. 3657, pp. 437–448.

[44] S. D. Servetto, C. I. Podilchuk, and K. Ramchandran, "Capacity issues in digital image watermarking," in *Proc. ICIP'98, IEEE Int. Conf. on Image Processing*, October 1998, pp. 445–449.

[45] M. U. Celik, G. Sharma, and A. M. Tekalp, "Collusion-resilient fingerprinting using random pre-warping," *IEEE Signal Processing Letters*, vol. 11, pp. 831–835, October 2004.

[46] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering - Fundamentals, Algorithms, and Standards*, Boca Raton, FL: CRC Press, 2003.

[47] W. H. Dobelle, "Artificial vision for the blind by connecting a television camera to the visual cortex," *ASAIO Journal*, vol. 46, pp. 3–9, 2000.

[48] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 997–1000, 2003.

[49] G. D. Cohen and H. G. Schaathun, "Asymptotic overview on separating codes," May 2003.

[50] A. D'yachkov and P. Vilenkin, "Families of finite sets in which no intersection of l sets is covered by the union of s others," *Journal of Combinatorial Theory*, vol. A, no. 99, pp. 195–218, 2002.

[51] D. Kundur and D. Hatzinakos, "Towards robust logo watermarking using multiresolution image fusion," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 185–198, February 2004.

APPENDIX A

ADDITIONAL MULTIMEDIA COLLUSION ATTACKS

From the estimation attack category, the optimal estimation in terms of the mean square error (MSE) criterion, given frames in $\{\tilde{C}_i\}_{i=1}^{M}$, is

$$\begin{aligned}\underline{C}^j(x,y) &= \mathbf{E}[C^j(x,y)|\tilde{C}_1^j(x,y),\tilde{C}_2^j(x,y),\ldots,\tilde{C}_M^j(x,y)]\\ &= h(\tilde{c}_1^j(x,y),\tilde{c}_2^j(x,y),\ldots,\tilde{c}_M^j(x,y)),\end{aligned}$$

where $\underline{C}^j(x,y)$ is an estimate of the j^{th} frame at the $(x,y)^{\text{th}}$ pixel of the respective frame and pixel in C (no fingerprint), and

$$h(\tilde{c}_1^j(x,y),\tilde{c}_2^j(x,y),\ldots,\tilde{c}_M^j(x,y)) \int_{-\infty}^{+\infty} z\, \frac{f_{C^j(x,y)\tilde{C}_1^j(x,y)\tilde{C}_2^j(x,y)\ldots\tilde{C}_M^j(x,y)}(z,\tilde{c}_1^j(x,y),\tilde{c}_2^j(x,y),\ldots,\tilde{c}_M^j(x,y))}{f_{\tilde{C}_1^j(x,y)\tilde{C}_2^j(x,y)\ldots\tilde{C}_M^j(x,y)}(\tilde{c}_1^j(x,y),\tilde{c}_2^j(x,y),\ldots,\tilde{c}_M^j(x,y))}\, dz \tag{A.1}$$

$f_{C^j(x,y)\tilde{C}_1^j(x,y)\tilde{C}_2^j(x,y)\ldots\tilde{C}_M^j(x,y)}$ is the joint probability density function (PDF) of the j^{th} frame, $(x,y)^{\text{th}}$ pixel random variables of C, and $\{\tilde{C}_i\}_{i=1}^{M}$. $f_{\tilde{C}_1^j(x,y)\tilde{C}_2^j(x,y)\ldots\tilde{C}_M^j(x,y)}$ is the joint PDF of the j^{th} frame, $(x,y)^{\text{th}}$ pixel random variables $\{\tilde{C}_i\}_{i=1}^{M}$.

Equation A.1 is a result from estimation theory. However, this estimation cannot be practically implemented, as possession of the PDFs is unlikely. Colluders must therefore resort to suboptimal estimation techniques.

Another linear attack that is more optimal than Equation 2.10 in the MSE sense, is given by Equation A.2.

$$\underline{C}^j(x,y) = \sum_{i=1}^{M} \alpha_i \left(\tilde{C}_i^j(x,y) - \mathbf{E}[\tilde{C}_i^j(x,y)]\right) + \mathbf{E}[C^j(x,y)] \tag{A.2}$$

The weights α_i in Equation A.2 are obtained by solving Equation A.3 for $k = 1, 2, \ldots M$.

$$\mathbf{E}\left[\left(C^j(x,y) - \mathbf{E}[C^j(x,y)] - \sum_{i=1}^{M} \alpha_i \left(\tilde{C}_i^j(x,y) - \mathbf{E}[\tilde{C}_i^j(x,y)]\right)\right)\left(\tilde{C}_k^j(x,y) - \mathbf{E}[\tilde{C}_k^j(x,y)]\right)\right] = 0 \tag{A.3}$$

Although Equation A.3 may seem intimidating, it is a linear equation of M equations (for $k = 1, 2, \ldots, M$) and M unknowns (α_i, for $i = 1, 2, \ldots, M$). The expected values are more easily estimated than the PDFs in Equation A.1.

Other suboptimal schemes may include adding $\{\tilde{C}^j(x, y)\}_{j=1}^{M}$, and applying an FIR filter as in Equation A.4.

$$\underline{C}^j = h(x, y) \circledast \sum_{i=1}^{M} \tilde{C}_i^j(x, y) \tag{A.4}$$

$h(x, y)$ is an FIR 2-D spatial filter, and $\circledast$ is the 2-D convolution operator. The goal of Equation A.4 is to attenuate the fingerprint by blurring the sum. This attack with AWGN, is termed the Gaussian Medium Access Channel (GMAC).

APPENDIX B

LIMITATIONS OF PREVIOUS CODES

A. Limitations of the Concatenated Separating Code

This section provides restrictions for the (t, t)-separating codes that are not mentioned in [19]. These findings will show that the concatenated (t, t)-separating codes are not suitable for larger coalition sizes, and the proposed novel code can easily outperform them in terms of codeword length.

First, the probability bound in Equation 3.7 can be simplified as provided in Equation B.1.

$$\epsilon \leq q^K 2^{-ND\left(\sigma \| \frac{t-1}{q-1}\right)} \tag{B.1}$$

Since the total codeword length is $n = mN$, a lower bound for both N and m will be derived in terms of only t and q, and hence the lower bound for the codeword length will also be in terms of t and q, providing an accurate comparison of the codeword length with that of other popular fingerprinting codes.

Equation B.1 is solved for N as provided in Equation B.2.

$$N \geq \frac{\log_2\left(\frac{q^K}{\epsilon}\right)}{D\left(\sigma \| \frac{t-1}{q-1}\right)} \tag{B.2}$$

To express Equation B.2 in terms of only ϵ, t, and q, it is noted that for linear Maximum Distance Separable (MDS) codes $N - K + 1 = \delta N$, since W is a linear $(N, K, \delta N)$ code, therefore substituting $K = (1 - \delta)N + 1$ into Equation B.2, and then solving for N again, results in Equation B.3.

$$N \geq \frac{\log_2\left(\frac{q}{\epsilon}\right)}{D\left(\sigma \| \frac{t-1}{q-1}\right) - (1-\delta)\log_2(q)} \tag{B.3}$$

Both $D\left(\sigma \| \frac{t-1}{q-1}\right)$ and δ are only dependent on t and q, so Equation B.3 only depends on t and q. The restrictions on δ are $\delta > 1 - \frac{1}{t^2} + \frac{t-1}{t(q-1)}$, but also $\delta < 1$, since $N > K = (1-\delta)N + 1 > 0$. This means that $\delta = 1 - \frac{1}{t^2} + \frac{t-1}{t(q-1)} + \varepsilon$, where ε must satisfy Equation B.4 in order to satisfy both restrictions on δ .

$$0 < \varepsilon < \frac{1}{t^2} - \frac{t-1}{t(q-1)} \tag{B.4}$$

The reader can check that using an ε that satisfies Equation B.4, will also result in satisfying the constraint that $\sigma \leq 1$. In addition $q \geq t$ is required (and makes sense since the size of V, being q, must be larger than its ability to (t,t)-separate) to satisfy the constraint $\frac{t-1}{q-1} \leq 1$. Most of these conditions are tacitly assumed in [19], however the next result presented in this thesis is probably overlooked in [19], as it shows the codeword length to increase exponentially in t.

In [19], the following inequality is given

$$\frac{\log_2(q)}{m} \geq \frac{-\log_2(1-2^{-(2t-1)})}{2t-1} - \frac{1}{m} \tag{B.5}$$

for (t,t)-separating codes. Solving for m and replacing the logarithm term by its Taylor series expansion, results in

$$m \leq \frac{(2t-1)(\log_2(q)+1)}{\sum_{n=1}^{\infty} \frac{2^{-(2t-1)n}}{n}} \leq 2^{(2t-1)}(2t-1)(\log_2(q)+1) = O(t2^t \log_2(q)) \tag{B.6}$$

Since the total length of the code is given by mN, and N does not contain any exponential terms, the total length is approximately exponential in t.

As further evidence, consider

$$\lim_{m\to\infty} \frac{\log_2(q)}{m} \leq \bar{R}(t,t),$$

where $\bar{R}(t,t)$ is the asymptotic rate bound. In [49], the asymptotic rate bounds for $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(5,5)$-separating codes are given as 1, 0.2835, 0.06627, 0.01630, 0.004037 respectively. This suggests that perhaps the asymptotic rate bound decreases exponentially in t when $m \to \infty$, so for large m, $m \approx \frac{\log_2(q)}{\bar{R}(t,t)}$ increases exponentially with t. The asymptotic rate bound in [49] is given as a recursive formula, so a close-formed bound is derived here. According to [49], the best asymptotic rate of a (t,t)-separating code is bounded by two times the best asymptotic rate of a (t,t)-superimposed code. For the purpose of this thesis, superimposed codes are not discussed, and only mentioned here to derive a closed-form expression of the rate bound for separating codes.

$$R^{\text{ separating-codes}}(t,u) \leq 2R^{\text{ superimposed-codes}}(t,u) \tag{B.7}$$

The rate bound for superimposed codes is also recursive in [49]. However [50] provides a closed-form asymptotic rate bound for superimposed codes.

$$\bar{R}^{\text{ superimposed-codes}}(t,u) \leq \frac{(t-1)^{t-1}(u-1)^{u-1}}{(t+u-2)^{t+u-2}} \tag{B.8}$$

Substituting Equation B.8 into B.7, and setting both t and u to t, the closed-form expression for the rate bound of separating codes is given in Equation B.9.

$$\bar{R}(t,t) \leq 2^{-(2t-3)} \tag{B.9}$$

As predicted, the asymptotic rate bound decreases exponentially in t.

Since we are interested in the codeword length for large t, and $m > t$, it might be reasonable to assume that for very large t and therefore very large m,

$m \approx 2^{2t-3}\log_2(q)$. The codeword length is approximately given by Equation B.10 for sufficiently large t.

$$\text{total length of codeword} > 2^{2t-3}\log_2(q)\frac{\log_2\left(\frac{q}{\epsilon}\right)}{D\left(\sigma\|\frac{t-1}{q-1}\right)-(1-\delta)\log_2(q)} \tag{B.10}$$

Although t and q appear elsewhere other than the power of 2 in Equation B.10, they do not have an exponential effect, hence the codeword length increases exponentially with t.

B. Limitations of the Erasable c-TA Codes

In [10], a c-TA code that can tolerate up to s erasures, using Reed-Solomon codes and the Guruswami-Sudan soft-decision decoder is presented. This section will show that the ability to tolerate a fixed number of erasures is not enough to qualify the code as being robust to the extended narrow-sense attack, and hence this code cannot be used for multimedia fingerprinting, as robustness against erasure is a must.

The linear MDS Reed-Solomon code is used with $d_{min} > n - \frac{n}{c^2} + \frac{s}{c^2}$, where n is the length of a codeword, s is the number of tolerable erasures, and c is the maximum coaltion size. Let $d_{min} = n - \frac{n}{c^2} + \frac{s}{c^2} + D$, where $D > 0$. This means that given any 2 codewords, there will be at least d_{min} positions that do not match (i.e. there are at least d_{min} detectable positions given any coalition). Given that the pirates know that a c-TA code is being used, the pirates' best strategy is to erase all detectable marks. This means that s, being the number of tolerable erasures, should be at least equal to d_{min}, or $s \geq n - \frac{n}{c^2} + \frac{s}{c^2} + D$, and solving for s, results in $s \geq n + \frac{D}{1-\frac{1}{c^2}}$.

At the same time, since the code is MDS, $d_{min} = n - k + 1$, which implies $n - \frac{n}{c^2} + \frac{s}{c^2} + D = n - k + 1$. Therefore $k = \frac{n-s}{c^2} - D + 1$. It is easy to see that if $s \geq n + \frac{D}{1-\frac{1}{c^2}}$, then $k < 0$, which cannot hold since $0 < k < n$. Hence s cannot be greater or equal to d_{min}, and any two pirates can erase more than s detectable bits, hence defeating this code.

APPENDIX C

MODULATION AND WATERMARKING FOR FINGERPRINTING CODES

This appendix pertains to details of modulation and watermarking tailored for fingerprinting codes. In particular, modulation and watermarking can be used to conceal fingerprinting codewords, and hence enhance the robustness of these codewords.

A. One-to-Many Modulation

Given a codebook Γ whose codewords are assembled from the alphabet Σ, a *one-to-many* modulation scheme can be applied to further conceal the codewords embedded in the media. Traditionally, the modulation function maps every element in Σ to one watermark. For example, given a binary alphabet, 0 might be modulated to watermark W_0, and 1 is modulated to watermark W_1. However, for fingerprinting codewords, 0 can be mapped to multiple watermarks, and 1 can also be mapped to multiple watermarks as is now described.

For every bit position, a different modulation function is applied. For example, for bit position 1, the modulation function might map 0 to $W_{0,1}$ and 1 to $W_{1,1}$. However for bit position 2, the modulation function can map 0 to $W_{0,2} \neq W_{0,1}$ and 1 to $W_{1,2} \neq W_{1,1}$. When a sequence of watermarks is extracted from the watermark extraction function, the demodulation function knows that the first watermark is demodulated according to $W_{0,1} \mapsto 0$ and $W_{1,1} \mapsto 1$, and the second watermark is demodulated according to $W_{0,2} \mapsto 0$ and $W_{1,2} \mapsto 1$. From this example, a binary codeword $\gamma^i = \gamma^i_1 \gamma^i_2 \cdots \gamma^i_l$ is modulated according to the bit-wise modulator:

$$\mathcal{M}(\gamma^i_k) = \begin{cases} W_{0,k} & \text{if } \gamma^i_k = 0 \\ W_{1,k} & \text{if } \gamma^i_k = 1 \end{cases} \tag{C.1}$$

Here $\mathcal{M}$ is a function from $\Sigma = \{0, 1\}$ to $\mathbb{W}$, which is different from the modulation function in Definition 9. In addition, $W_{0,k}$ and $W_{1,k}$ must be unique, however it is possible to have $W_{0,k} = W_{1,j}$ for $k \neq j$, which can confuse the attackers.

B. Erasure in Multimedia

For extended narrow-sense or wide-sense attacks, the detectable bit can be "erased" or mapped to any symbol that is not in Σ. This section shows that for multimedia, erasure is restricted.

In any watermarking algorithm, a transform coefficient of an image can only be changed by an amount that will not affect the visual quality of the resulting image [1, 22, 51]. Therefore if a coalition of pirates detects differences in a set of coefficients, the amount of change that they can apply to these coefficients is also limited. Figure 28 shows the effects of randomly erasing 1% of the DCT coefficients in an image, by setting these coefficients to 0. Distortion is evident in the background of Figure 28(b). Figure 29 shows the effect of randomly erasing 6% of the db2 wavelet coefficients, by

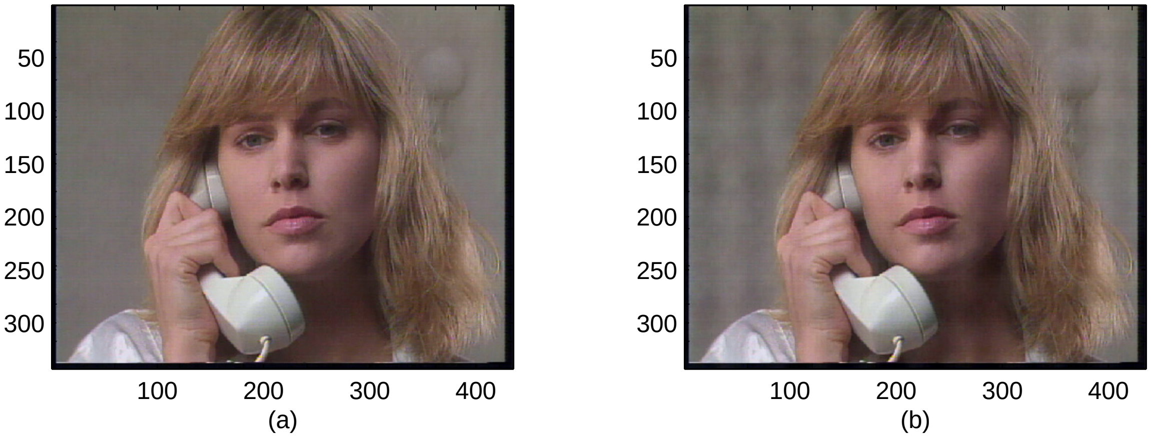


Fig. 28. (a) Original Image; (b) Image After 1% (Random) of the DCT Coefficients Are Set to 0

setting these coefficients to 0. Distortion is evident in the grainy edges and lips of Figure 29(b).

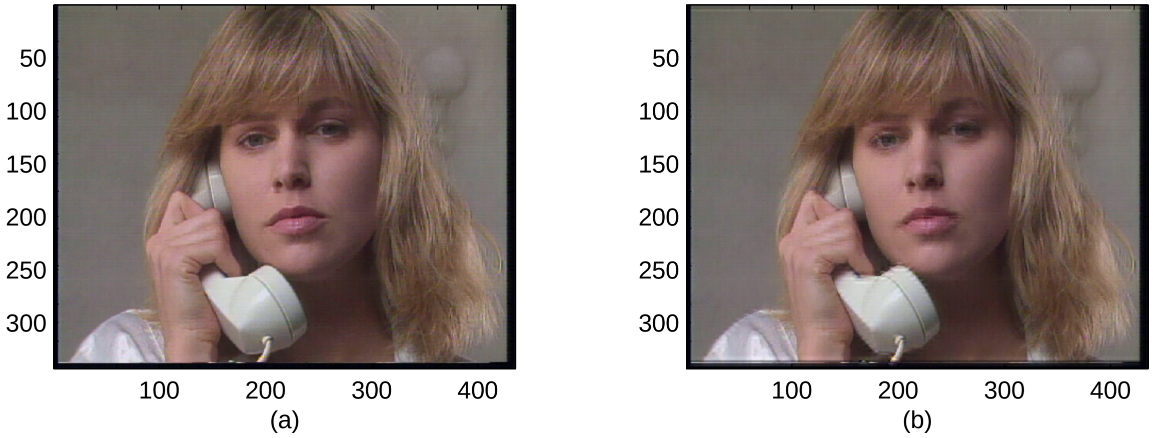


Fig. 29. (a) Original Image; (b) Image After 6% (Random) of the db2 Wavelet Coefficients Are Set to 0

In general, when a coalition of pirates detects differences in transform coefficients, they can only alter these coefficients by a certain amount. Therefore the modulation function can even modulated a 0 in a fixed bit position, for example, to *different* watermarks that share some similarities with one another. If a coalition of pirates detects these different watermarks all representing 0, the amount of altering that they can achieve without visual distortion will not be enough to fool the demodulation function, which will still detect a 0 after the watermarks have been "erased". The modulation function is then given by:

$$\mathcal{M}(\gamma_k^i) = \begin{cases} W_{0,i,k} & \text{if } \gamma_k^i = 0 \\ W_{1,i,k} & \text{if } \gamma_k^i = 1 \end{cases} \tag{C.2}$$

where $W_{0,i,k}$'s are "similar" for all varying i and fixed k, and $W_{1,i,k}$'s are "similar" for all varying i and fixed k. The design and implementation of such modulators and demodulators are beyond the scope of this thesis.

APPENDIX D

DETAILS PERTAINING TO THE PROPOSED SUB-OPTIMAL JSF ALGORITHM

A. Justification of Using the Minimum Average Motion Distance Detector

As in the body of this thesis, let $\Phi_i = (\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n})$ be a set of frames corresponding to the fingerprints for User i. Let $\tilde{\Phi} = (\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_n)$ be the set of frames created by a collusion attack as defined by the body of this thesis. The soft-decision maximum likelihood detector will choose one suspect based on Equation D.1.

$$\text{user } i \text{ is a suspect} = \arg\max_i Pr[\widetilde{\Phi} \mid \Phi_i] \tag{D.1}$$

The following set of equations will justify the use of the minimum average motion distance. Many assumptions need to be made, because the probability density function of $Pr[\widetilde{\Phi} \mid \Phi_i]$ is unknown and can never be estimated if the collusion attack is unrestricted. It will be shown that these assumptions are better than just assuming a popular distribution, such as the Gaussian distribution.

$$\begin{aligned}
Pr[\widetilde{\Phi} \mid \Phi_i] &= Pr[\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_n \mid \phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}] && \text{(D.2)} \\
&= \prod_{j=1}^{n} Pr[\tilde{\phi}_j \mid \phi_{i,j}] && \text{(D.3)} \\
&= \prod_{j=1}^{n} h(\tilde{\phi}_j) \exp(-AM(\tilde{\phi}_j, \phi_{i,j})) && \text{(D.4)} \\
&= \left(\prod_{j=1}^{n} h(\tilde{\phi}_j) \right) \exp\left(-\sum_{k=1}^{n} AM(\tilde{\phi}_k, \phi_{i,k}) \right) && \text{(D.5)}
\end{aligned}$$

$$\therefore \arg\max_i Pr[\widetilde{\Phi} \mid \Phi_i] = \arg\min_i \sum_{k=1}^{n} AM(\tilde{\phi}_k, \phi_{i,k}) \tag{D.6}$$

Equation D.3 is arrived at by assuming $\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}$ are independent, $\tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_n$ are independent, and $\tilde{\phi}_i$ is independent of ϕ_j for all $j \neq i$. The first independence assumption holds if the frames $\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}$ are not consecutive, and spaced out far enough - if these conditions hold, then the second assumption will automatically hold, because collusion will only occur with frames that are somewhat close to one another. In practice, this assumption does not always hold, since the frames $\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,n}$ are probably correlated.

Equation D.4 first assumes that the average motion increases when frames become further apart. In general, this assumption is valid, unless moving objects in the frames return to a position that occured in a previous frame. Given that this assumption is true, assume that frame $\phi_{i,j}$ is indeed part of a fingerprint from the coalition. Then any attacked frame generated with $\phi_{i,j}$ being part of the collusion attack will in general be a function of the frames that are close to $\phi_{i,j}$. This means that the probability that $AM(\tilde{\phi}_j, \phi_{i,j})$ is small, is very high, while the probability that $AM(\tilde{\phi}_j, \phi_{i,j})$ is large, is very small. Therefore, given $\phi_{i,j}$ is part of the collusion attack, $Pr[\tilde{\phi}_j \mid \phi_{i,j}]$ is a decreasing function of $AM(\tilde{\phi}_j, \phi_{i,j})$. In order to get a nice closed-form expression, it is assumed that this function is a decreasing exponential. The other equations follow from Equation D.4.

These assumptions made in deriving the minimum average motion detector are actually less stringent than just assuming a popular probability density function for $Pr[\tilde{\phi}_j \mid \phi_{i,j}]$. For example, the Gaussian PDF is usually assumed, however this assumption leads to the minimum Euclidean distance detector, which cannot be used as pointed out in Chapter V. In addition, the simulations are run on the weakest input as mentioned in Chapter V, but still give good results. Most of the frames in the test input video are highly correlated with one another, since there is not much motion, so the first independence assumption is violated; even so, the performance of this detector is far better than the minimum Euclidean distance detector, which almost always failed at correctly identifying a suspect.

B. Proof of Equation 5.22 for Supporting a Larger Set of Users

Let n be the original number of fingerprints, and $N \gg n$ be the new number of fingerprints. Let L be the number of partitions, and $c < \frac{n}{2}$ be the maximum size of the coalition. Given a word $\tilde{w}$, at least $\frac{L}{c}$ positions will match with one of the pristine codewords $w_i \in W$ of the attackers. This is based on the pigeonhole principle, where each of the original c codewords is analogous to c pigeonholes, while the L partitions represent L pigeons; at least one pigeonhole will contain at least $\frac{L}{c}$ pigeons. The proof will show that random words that are not part of the coalition of attackers will match in at least $\frac{L}{c}$ positions with the probability approximation presented in Equation 5.22.

The probability that a random codeword matching in x positions to $\tilde{w}$ is described by a binomial random variable X, with parameters $(L, \frac{1}{n})$, where the success probability of one position matching is $\frac{1}{n}$, since the alphabet size is n. X can also be approximated by a Gaussian random variable Y, with the same mean $\frac{L}{n}$ and variance $\frac{L(n-1)}{n^2}$, since L is typically very large, while n is about 10.

$$P\left(X \geq \frac{L}{c}\right) \approx P\left(Y \geq \frac{L}{c}\right) \tag{D.7}$$

$$= Q\left(\sqrt{\frac{L}{n-1}}\left(\frac{n}{c} - 1\right)\right) \tag{D.8}$$

Since there are N codewords in total, the union bound is then given by Equation 5.22, completing the proof.

APPENDIX E

ADDITIONAL SIMULATION RESULTS FOR THE JSF ALGORITHM

This appendix provides additional simulation results to those presented in Chapter V. The test video in this section is fundamentally different from the test video used in Chapter V, in that the motion in this test video is a zooming motion. Although the translation model is not accurate for this video, the simulation results show that the proposed algorithm still correctly identifies one pirate, and also provides visual degradation against collusion attacks.

In Figure 30, the single-user attacks do not mimic motion, hence cannot fool the detector. In Figures 31 and 33, one pirate is always correctly identified. In Figures 32 and 34, the attacked frames from the proposed algorithm incur visual distortion, while the attacked frames from the DCT-based (Figure 32(c)) and wavelet-based (Figure 34(c)) watermarking techniques [1, 22] are perceptually similar to the original frames.

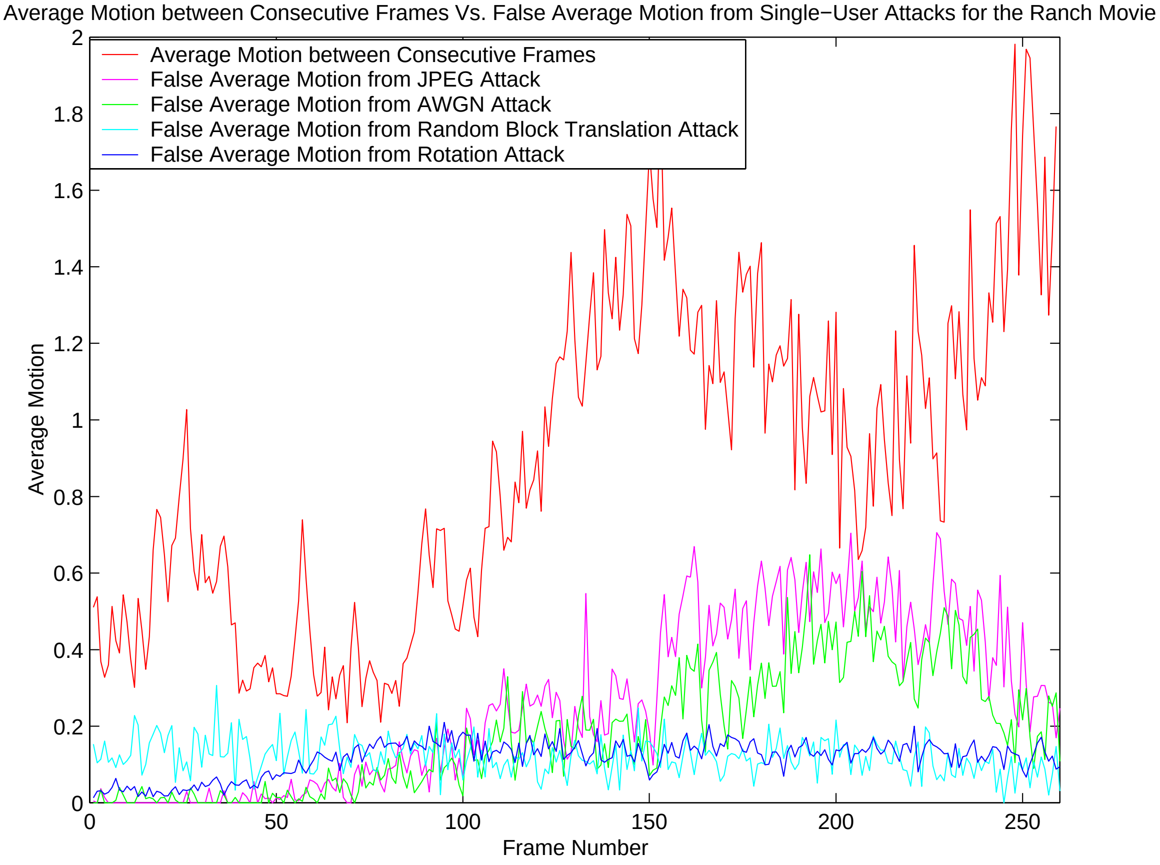


Fig. 30. Average Motion Between Consecutive Frames and False Average Motion from Single-User Attacks

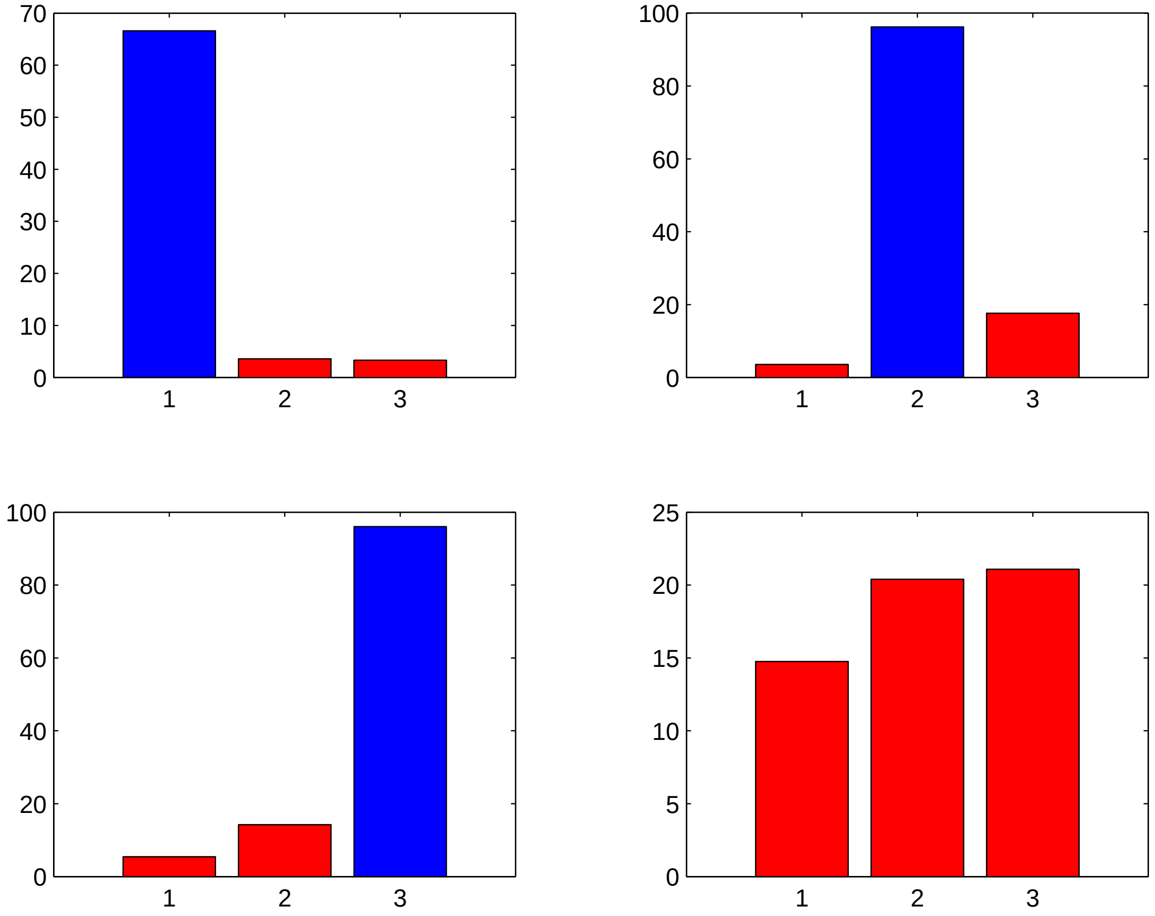


Fig. 31. Y-axis on Each Graph Is the Average Motion Distance Between the Average Attacked Video (Averaging of Fingerprinted Videos Whose Bars Are Red) and the Fingerprinted Video (Whose Number Is on the X-axis)

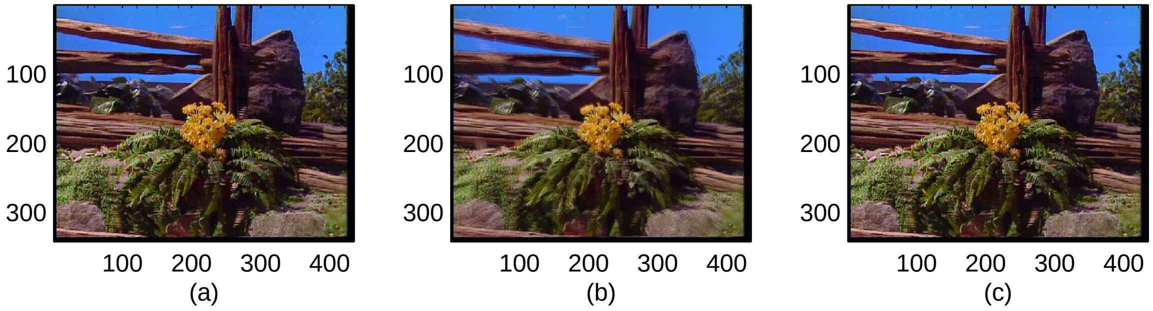


Fig. 32. (a) Original Frame; (b) Blurry Frame After Average Attack; (c) Average Attack on 60 Watermarked Frames

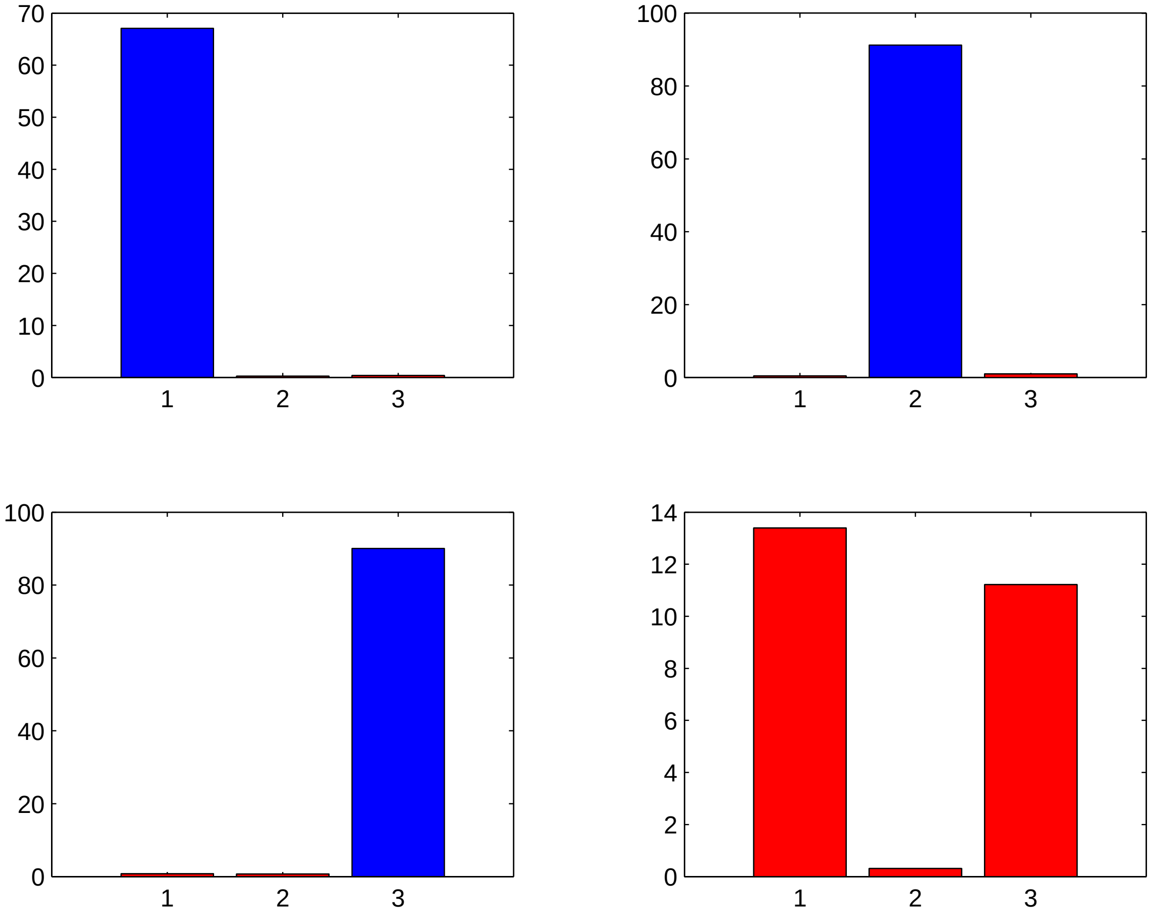


Fig. 33. Y-axis on Each Graph Is the Average Motion Distance Between the Random Scrambling Attacked Video (Random Scrambling of Fingerprinted Videos Whose Bars Are Red) and the Fingerprinted Video (Whose Number Is on the X-axis)

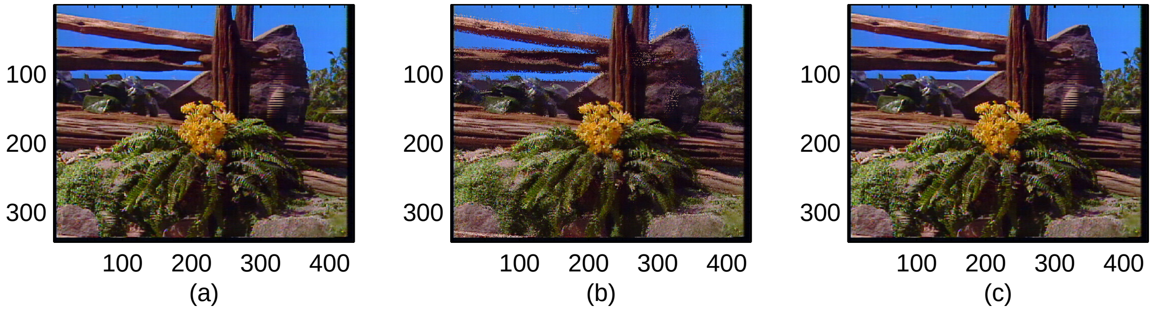


Fig. 34. (a) Original Frame; (b) Distortion After a Random Scrambling Attack; (c) Random Scrambling on 60 Watermarked Frames

VITA

William Luh was born in Mississauga, Ontario, Canada in 1979. He received his B.A. in computer engineering from the University of Toronto, Toronto, Ontario in 2002. During the summers of 2000 to 2002, he worked at General Electric's Electromagnetics Laboratory in Peterborough, Ontario, and A.U.G. Signals Ltd. in Toronto, Ontario, where he was a software developer and research assistant. Since 2003, he has been a Research Assistant and Graduate Assistant in the Department of Electrical Engineering at Texas A&M University. His research interests include multimedia security, digital rights management, and signal processing for multimedia. His permanent address is: 4538 Mayflower Drive, Mississauga, Ontario, L5R 1S3, Canada.

The typist for this thesis was William Luh.