# FINGERPRINTING CODES: HIGHER RATES, QUICK ACCUSATIONS

by

Ehsan Amiri

B.Sc., Sharif University of Technology, Tehran, Iran, 2001

M.Sc., Sharif University of Technology, Tehran, Iran, 2003

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science


SIMON FRASER UNIVERSITY
Fall 2010

## APPROVAL

**Name:** Ehsan Amiri

**Degree:** Doctor of Philosophy

**Title of thesis:** Fingerprinting Codes: Higher Rates, Quick Accusations

**Examining Committee:** Dr. Valentine Kabanets
Chair

Dr. Gabor Tardos, Senior Supervisor

Dr. Andrei Bulatov, Supervisor

Dr. Valentine Kabanets, Supervisor

Dr. David Mitchell, Supervisor

Dr. Funda Ergun, SFU Examiner

Dr. Rei Safavi-Naeini, External Examiner,
Professor of Computer Science,
University of Calgary

**Date Approved:** October 1, 2019

# Declaration of
# Partial Copyright Licence

# Abstract

Leakage of a document distributed among a set of authorized recipients may result in violation of copyright or secrecy of a document. A hidden serial number in the document can be used for finding the source of the leak. A coalition attack is collusion of a set of users to generate copies of the document with a serial number (now called fingerprint) that is different from their original ones. To trace such a forged fingerprint back to one of its producers, reasonable assumptions are necessary. The marking assumption is one such assumption. It says if the $i$-th bits of the fingerprints of all colluders are the same, it will have the same value in the forged copy.

Our first result in this thesis is a construction that achieves the highest known rate of fingerprinting codes, which we conjecture to be optimal [3]. This construction combines two ideas from two earlier constructions. We use bias based code generation that was introduced by Tardos [44] for his well-known fingerprinting code. Our accusation algorithm is based on an earlier algorithm by Anthapadmanabhan, Barg and Dumer [7] that uses the information theoretic notion of typicality. The drawback of this construction is its slow accusation algorithm.

Building upon our first construction, we construct another fingerprinting code in which the distributor may choose to have a faster accusation algorithm by sacrificing a little of the rate. A different accusation algorithm for our first code allows us to generalize it to a family of codes that show a tradeoff between rate and efficiency.

This tradeoff suggests new ways to construct more efficient algorithms without losing the rate. For the case of two pirates this tradeoff can be made simpler by using Hamming distance instead of a mutual information. This allows improving quadratic running time of our first accusation algorithm to linear without lowering the rate at all.

We also look at weak fingerprinting, a variant of fingerprinting for which the capacity is

known. We construct a capacity-achieving weak fingerprinting code.

**Keywords:** fingerprinting; collusion attack; rate; capacity; mutual information

# Acknowledgments

It has been a great pleasure for me to work with Gabor Tardos during my PhD. Gabor was always surprising me with his profound insight and unexpected attention to details. It was amazing to see how often he finds a mistake in a carefully reviewed proof, sometimes in published ones. His patience and welcomness was a great source of support for me.

I would like to thank Drs. Andrei Bulatov, Funda Ergun, Valentine Kabanets, David Mitchell, and Rei Safavi Naeini for reviewing the thesis and attending the defence. I also thank Valentine for the work that I did with him earlier in my PhD. I thank all my friends at SFU who made my years in Vancouver even more memorable.

I cannot express my appreciation for my parents' patience and support during my long PhD career. In the last year of my PhD I married Laleh, my dear wife, which made the last year very different from earlier ones. I dedicate this thesis to my parents and Laleh.

# Contents

**Approval** **ii**

**Abstract** **iii**

**Acknowledgments** **v**

**Contents** **vi**

**1 Introduction** **1**

1.1 Other research . . . . . 2

**2 Tools** **4**

2.1 Information theory . . . . . 4

2.2 Basic concepts . . . . . 4

2.3 Method of types . . . . . 8

2.3.1 Conditional types . . . . . 10

2.4 Game Theory . . . . . 11

2.4.1 Minimax theorem . . . . . 13

2.4.2 Infinite games . . . . . 14

**3 Context** **16**

3.1 Traitor Tracing . . . . . 17

3.2 Fingerprinting . . . . . 19

3.2.1 The model . . . . . 20

3.2.2 Lower bounds on fingerprinting capacity . . . . . 22

3.2.3 Upper bounds on fingerprinting capacity . . . . . 25

3.2.4 Codes with identifiable parent property . . . 25
3.3 Steganography . . . 26
3.3.1 Theoretical questions . . . 28

**4 Higher Rates** **29**
4.1 Basics . . . 30
4.1.1 Bias Based Code Generation . . . 30
4.1.2 Channel based pirate strategies . . . 30
4.2 The fingerprint game . . . 31
4.2.1 Details of construction . . . 32
4.3 Proofs . . . 34
4.4 Numerical Results . . . 37
4.4.1 Proof of Lemma 13 . . . 38
4.5 Estimating $R_t$ . . . 39
4.6 Support of $D^t$ . . . 44

**5 Quick Accusations** **47**
5.1 Modifying the accusation algorithm . . . 48
5.1.1 The tradeoff . . . 53
5.2 A direction for improvement? . . . 55
5.3 A fast accusation algorithm for two pirates . . . 57
5.3.1 A useful lemma . . . 58
5.3.2 Accusation algorithm . . . 58
5.4 Weak fingerprinting . . . 59

**6 Open problems** **62**
6.1 Capacity . . . 62
6.2 Better constructions . . . 63
6.3 Better evaluation of rates . . . 64

# Chapter 1

# Introduction

This thesis is focused on fingerprinting under marking assumption. The main purpose of fingerprinting codes is protecting copyright and/or secrecy of a document.

Fingerprinting was born as a result of research on traitor tracing codes. Traitor tracing research started in mid 90's and so far it has resulted in a few different and independent research problems. In the Chapter we will briefly review traitor tracing and some related problems.

Fingerprinting codes have two main components. One is the code-matrix, which is the set of codewords assigned to users. The other is the accusation algorithm. One line of fingerprinting research tries to construct better codes by decreasing code length or time complexity of accusation algorithms. Our contributions to fingerprinting research, which are presented in this thesis, are all in this line of research as we will see in Chapters 4 and 5. There is another line of research that tries to prove lower bounds on the code length of fingerprinting codes. We do not consider this problem in this thesis.

As we will see soon, fingerprinting without randomization is impossible, probabilistic analysis tools are required for working on fingerprinting. More recent results, including results in this thesis, have used tools and concepts from information theory and game theory.

Chapter 2 of this thesis summarizes information theoretic and game theoretic tools needed in the rest of the thesis.

Chapter 3 has two roles. First it presents a big picture of the research area to which fingerprinting belongs. Second, it introduces fingerprinting under marking assumption by presenting definition and a quick review of main results on the problem.

Our new results on fingerprinting are presented in Chapters 4 and 5. While results of Chapter 4 and the last section of Chapter 5 have been published in [3], most of the results in Chapter 5 are not yet submitted and will be submitted for publication.

Chapter 4 presents our construction of high rate fingerprinting codes. In this chapter we construct our first fingerprinting codes. This code is the shortest fingerprinting code that we are aware of. We conjecture that it achieves the shortest possible length. This code is also the basis of our other constructions.

Informally speaking the construction of Chapter 4 achieves the shortest fingerprinting code, but its accusation algorithm is not very fast. In Chapter 5 we address this problem by introducing new constructions with faster accusation algorithms.

The first construction in this chapter is a fingerprinting code with a tradeoff parameter. This parameter can be tuned during the construction such that the outcome of the construction is a code with faster accusation algorithm. The price that we need to pay is to sacrifice a little bit of the rate. As we will see the rate that we lose is not significant.

In the special case of two pirates we are able to improve the construction so that we achieve faster fingerprinting code without sacrificing the rate.

Also in this chapter we present our result on weak fingerprinting. This is a different model of fingerprinting which was presented in [3] first.

In Chapter 4 we introduce the notion of fingerprinting game. This concept underlies all constructions presented in this thesis.

Finally we conclude by looking at open problems and research in progress.

## 1.1 Other research

The author has conducted research on different problems, during his PhD studies. The focus of this thesis is on fingerprinting research.

Another research problem is the analysis of SAT heuristics. The aim of this line of research is to understand why some simple heuristic algorithms are successful in quickly solving random instances of hard problems (in this case SAT).

In particular in [2] we have focused on the so called Planted SAT and introduced a variation of random walk heuristic that works surprisingly quickly on planted SAT. The original random walk heuristic with high probability requires exponential time to solve instances of planted SAT with densities larger than a constant [6]. Our result on our variant

of random walk [2] is partly experimental and partly theoretical. In the experimental part we show that our algorithm finds a solution for instances of planted SAT of any density. This is interesting in comparison to other heuristics designed for planted SAT like [22, 23]. These heuristics work for planted SAT with large enough constant density while our algorithm finds a solution for instances of planted SAT of any arbitrary density.

In the theoretical part we have shown that this algorithm almost surely finds the satisfying assignment of a full 3-CNF formula with a planted solution, in time linear in the number of variables. Also we show that for a 3-CNF formula with a planted solution $r$ and super constant density,[1] with high probability the algorithm finds a vector in which a fraction $\leq \epsilon$ of variables differ from $r$. Here $\epsilon$ is a constant. Notice that the proof does not imply that this vector is a satisfying assignment. What makes this result valuable is that we know that for high density planted SAT instances all satisfying assignments have small Hamming distance from the planted solution.

The author has also conducted research in a significantly different area of computer science. As a member of COSTAR lab at Simon Fraser University the author has been involved in the development of Parallel Bit Streams Technology [14, 17, 15]. The aim of this technology is to increase the performance of text processing tasks (such as parsing and transcoding) through parallelization. We exploit advanced features of modern processors for parallelization. We have been successful in sharply increasing performance of transcoding [14] and parsing[15].

Developing software based on this technology is a tedious and time consuming task as it requires low level programming. So, in addition to development of Parallel Bit Streams techniques, we have designed a domain-specific language that is aware of Parallel Bit Streams. An optimizing compiler has been developed that given an input program that is written in our domain-specific langauge, produces optimized code in C. The XML parser reported in [15] is developed using this technology.

[1] Here we mean the density should go to infinity when number of variables goes to infinity.

# Chapter 2

# Tools

Throughout this thesis we will use tools from calculus, probability theory, information theory and game theory. Usually students of computer science take courses in calculus and probability theory. In order to make the content of this thesis accessible to computer science students who may be interested in the topic, in this chapter we cover tools from information theory and game theory that are needed for the rest of this thesis. Theorems that are crucial for the results are stated with a proof, even though they might be standard. Some easier or less crucial proofs are omitted.

## 2.1 Information theory

In this section we review concepts and tools from information theory that will be used throughout this thesis. For an in depth study of these concepts the reader is referred to [18, 20], even though [20] requires a more solid background in mathematics than [18].

## 2.2 Basic concepts

*Binary entropy*, or *entropy* for short, of a distribution $P$ over a countable set $\mathcal{X}$ is defined as

$$H(P) = \sum_{x \in \mathcal{X}} -P(x) \log P(x),$$

where log denotes binary logarithm. By convention we assume that if $P(x) = 0$, then $P(x) \log P(x) = 0$. Informally entropy shows the unpredictability of the random experiment of drawing an element of set $\mathcal{X}$ with respect to distribution $P$. A good example to demonstrate this is when $P$ is a binary distribution, i.e. $\mathcal{X} = \{0, 1\}$. Since this is an important case we use the following notation for it. Let $p = P(0)$, then $h(p) = h(1 - p) = H(P)$. Now the reader may easily verify that $h(0) = h(1) = 0$. In these two cases the value of a random variable drawn from the distribution is completely predictable, consequently the entropy is zero. Uniform binary distribution is the most unpredictable binary distribution in an informal sense. The notion of entropy formalizes this intuition as for any $p \in [0, 1]$, $h(p) \leq h(1/2) = 1$.

For a random variable $X$ that has distribution $P$ we define $H(X) = H(P)$.

*Conditional entropy* of two random variables $X$, $Y$ is defined as

$$H(X|Y) = \sum_y Pr(Y = y)H(X|Y = y).$$

$Pr(Y = y)$ denotes the probability that random variables $Y$ takes value $y$. If $Pr(Y = y) = 0$ we let $Pr(Y = y)H(X|Y = y) = 0$ by convention. $H(X|Y = y)$ is the entropy of $X$ after fixing $Y = y$. To calculate this we consider the conditional probability space obtained by fixing $Y = y$. Then let $\hat{P}$ denote the distribution of $X$ in this conditional probability space. Now $H(X|Y = y) = H(\hat{P})$. For example if $X \in \{1, 2, 3, 4, 5, 6\}$ is the result of throwing a die and $Y \in \{0, 1\}$ is a binary variable that is one if the result of throwing the die is divisible by three, then $H(X) = \log 6$, while $H(X|Y = 1) = 1$ and $H(X|Y) = 5/3$.

Alternatively we can define conditional entropy using the notion of joint entropy. If $X$ and $Y$ are two random variables taking values in $\mathcal{X}, \mathcal{Y}$, then we can define $Z = (X, Y)$ to be a random variable that takes values in the set $\mathcal{X} \times \mathcal{Y}$. Then we define $H(X, Y) = H(Z)$. Now we have,

$$H(X|Y) = H(X, Y) - H(Y). \tag{2.1}$$

We leave it to the reader to verify that these two alternative definitions of conditional entropy are equivalent.

Informally speaking, $H(X|Y)$ indicates unpredictability of $X$ if the value of $Y$ is known. Naturally this results in defining a new concept known as *mutual information* of two random variables denoted by $I(X; Y)$ and defined as

$$I(X; Y) = H(X) - H(X|Y).$$

$I(X;Y)$ tells us the decrease in unpredictability of $X$ if value of $Y$ is known. Using the second definition of conditional entropy one can show $I(X;Y) = I(Y;X)$.

The reader may expect that $H(X|Y) \leq H(X)$. This is indeed true. We skip the proof which is based on concavity of the log function. As a conclusion we have $I(X;Y) \geq 0$. Entropy is a concave function. We will use the following inequality later. For $p, q, \alpha \in [0,1]$

$$\alpha h(p) + (1-\alpha)h(q) \leq h(\alpha p + (1-\alpha)q). \tag{2.2}$$

Another important concept in information theory is *relative entropy* or *Kullback-Leibler divergence*. It is also known as *KL divergence* for the purpose of brevity. Assume $\mathcal{P}_1$ and $\mathcal{P}_2$ are two distributions on the same countable set $S$. Then

$$D(\mathcal{P}_1||\mathcal{P}_2) = \sum_{i \in S} \mathcal{P}_1(i) \log \frac{\mathcal{P}_1(i)}{\mathcal{P}_2(i)}.$$

In all cases that $\mathcal{P}_2(i) = 0$ we let the corresponding summand is equal be zero by convention. When $\mathcal{P}_1$ and $\mathcal{P}_2$ are binary distributions, let $p = \mathcal{P}_1(0)$ and $q = \mathcal{P}_2(0)$. Then the value of $D(\mathcal{P}_1||\mathcal{P}_2)$ is denoted by $h(p||q)$.

To gain intuition about relative entropy, assume we are generating random samples from a set according to one of the two distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ and we want to guess which distribution is being used for generating samples. The larger the relative entropy is the easier it is to realize that samples are from $\mathcal{P}_1$. As an extreme case $D(\mathcal{P}_1||\mathcal{P}_1) = 0$. There is a beautiful and somewhat surprising connection between mutual information of two variables and their relative entropy. Assume random variables $X$ and $Y$ are distributed according to distributions $\mathcal{P}_X$ and $\mathcal{P}_Y$ and let $\mathcal{P}_{XY}$ denotes their joint distribution. Then

$$I(X;Y) = D(\mathcal{P}_{XY}||\mathcal{P}_X\mathcal{P}_Y),$$

i.e. the mutual information of two random variables is equal to the relative entropy of the corresponding joint distribution and product distribution.

**Pinsker's inequality**

It is not hard to guess that having relative entropy in a formula makes it hard to manipulate the formula. Pinsker's inequality is a useful tool that sometimes can be used to estimate the formulas involving relative entropy. We need the following definition.

**Definition.** For two distributions $P_1$ and $P_2$ over the same countable set $S$, the $L_1$ distance of $P_1$ and $P_2$ is denoted by $||P_1 - P_2||$ and defined as

$$||P_1 - P_2|| = \sum_{a \in S} |P_1(a) - P_2(a)|$$

In this thesis, we only use Pinsker's inequality for binary distributions. Here we state the lemma in its general form, but we skip the complete proof and only give the proof of the binary case.

**Lemma 1**

$$D(P_1||P_2) \geq \frac{1}{2\ln 2}||P_1 - P_2||^2.$$

**Proof [18]:** Assume $P_1$ and $P_2$ are binary distributions with parameters $p$ and $q$ and $p \geq q$. Let

$$g(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q} - \frac{2}{\ln 2}(p-q)^2.$$

Then

$$\frac{dg(p,q)}{dq} = \frac{q-p}{\ln(2)(1-q)q} - \frac{4}{\ln 2}(q-p).$$

Since $q(1-q) \leq 1/4$ and $p \geq q$ we conclude that the above derivative is negative. By increasing $q$ the value of $g(p,q)$ decreases until $g(p,q) = 0$ for $p = q$. This finishes the proof of the binary case. ■

The following stronger form of Pinsker's inequality for binary distributions is taken from [3].

**Lemma 2** *For $p \leq q \leq 1/2$ or $p \geq q \geq 1/2$ we have,*

$$h(p||q) \geq \frac{(p-q)^2}{2\ln(2)q(1-q)}$$

$$h(q||p) \geq \frac{(p-q)^2}{2\ln(2)q(1-q)}$$

**Proof:** Consider the function $g(p,q) = h(p||q) - \alpha(p-q)^2$ for an arbitrary real $\alpha$. For $p = q$, $g(p,q) = 0$. Similar to lemma 1 we look at $\frac{dg(p,q)}{dq}$. If $\alpha < 1/2\ln(2)q(1-q)$ this derivative is negative when $q < p$ and positive when $p > q$. If we furthermore restrict the values of $p$ and $q$ to $p \leq q \leq 1/2$ or $p \geq q \geq 1/2$ we can guarantee that the value of the derivative has the same sign for all values between $p$ and $q$. In either case, this implies the first inequality claimed. The second one has a similar proof.■

We will also use the notion of conditional relative entropy. For distributions $V$ and $W$ we define

$$D(V||W|Z) = \sum_z Pr[Z = z]D(V_z||W_z). \tag{2.3}$$

Here $V_z$ and $W_z$ denote marginal distributions of $V$ and $W$ after fixing $Z$ to $z$.

## 2.3 Method of types

Method of types is a combinatorial tool in information theory. It has been used in the information hiding literature. A presentation on application of this method in information hiding can be found at [48]. The method of types is used frequently in Chapter 4 and 5 of this thesis. The reader may refer to [20] or [18] for a more in depth treatment of the subject. Also [19] is an independent tutorial on the method of types.

Assume that $U = X_1X_2....X_n$ is a string where the $X_i$ are drawn from an alphabet $\Sigma = \{1, 2, ..., c\}$. Let $N(i|U)$ be the number of occurrences of symbol $i$ in $U$. Then the type of $U$, denoted by $T$, is a $c$-tuple of rational numbers $T = (\frac{N(1|U)}{n}, ..., \frac{N(c|U)}{n})$.

**Representations of a type.** According to the above definition the standard representation of a type is a tuple of rational numbers, the sum of which is one. We may refer to an individual coordinate of this tuple by $T(a)$ for $a \in \Sigma$. An important alternate view of a type is a probability distribution over the underlying alphabet. For example the type $T$ that we just defined is a probability distribution over alphabet $\Sigma$. These two alternative representations of a type will be used as needed.

For each $i$, $N(i|U)$ may have $n + 1$ different values so the number of types of a string of length $n$ is at most $(n + 1)^c$. This result is useful mostly because the upper bound of $(n + 1)^c$ on the number of types is small (polynomial in $n$), compared to $c^n$, the number of all strings, which is exponential in $n$.

**Lemma 3** *Let $P_{c,n}$ be the number of types of strings of length $n$ drawn from a size $c$ alphabet. Then*

$$P_{c,n} \leq (n + 1)^c$$

In a setting where strings are generated randomly, it is interesting to ask what is the probability of obtaining a string of a given type. Not surprisingly if symbols in a string are independent and identically distributed, then all strings in a type class are of the same

probability. The following theorem will be used later to prove some results on types. Also it establishes a connection between entropy and relative entropy.

**Theorem 4** *Let $X_1, X_2, ..., X_n$ be chosen independently from an alphabet $\Sigma$ according to a distribution $\mathcal{Q}$ on $\Sigma$. Then for a given sequence $U$, the probability of $U = X_1X_2...X_n$ depends only on the type of $U$ and is equal to*

$$2^{-n(H(T_U)+D(T_U||\mathcal{Q}))}$$

*Here $T_U$ denotes type of $U$ and is considered as a distribution on the alphabet as well.*

**Proof:** Let $U = U_1U_2...U_n$. Then

$$\begin{aligned}
Pr(U = X_1X_2...X_n) &= \prod_{i=1}^{n} Pr(X_i = U_i) && (2.4)\\
&= \prod_{a\in\Sigma} (\mathcal{Q}(a))^{N(a|U)} && (2.5)\\
&= \prod_{a\in\Sigma} (\mathcal{Q}(a))^{nT_U(a)} && (2.6)\\
&= \prod_{a\in\Sigma} 2^{nT_U(a)\log \mathcal{Q}(a)} && (2.7)\\
&= \prod_{a\in\Sigma} 2^{nT_U(a)\log \mathcal{Q}(a) - T_U(a)\log T_U(a) + T_U(a)\log T_U(a)} && (2.8)\\
&= 2^{n\sum_{a\in\Sigma} -T_U(a)\log \frac{T_U(a)}{\mathcal{Q}(a)} + T_U(a)\log T_U(a)} && (2.9)\\
&= 2^{-n(H(T_U)+D(T_U||\mathcal{Q}))}. \blacksquare && (2.10)
\end{aligned}$$

Consequently if the type of $U$ is $\mathcal{Q}$, then $Pr(U) = 2^{-nH(T_U)}$.

The set of all sequences that have the same type $T$ is called a *type class* and is denoted by $C(T)$.

**Theorem 5** *For a type class $C(T)$,*

$$\frac{2^{nH(T)}}{(n+1)^c} \leq |C(T)| \leq 2^{nH(T)}$$

**Proof:** First we prove the upper bound. Assume we draw $n$ symbols from alphabet $\sigma$ according to the distribution $T$. Then theorem 4 implies that the probability of obtaining a string of type $T$ is $|C(T)|2^{-nH(T)} \leq 1$. This implies the upper bound.

For the lower bound we show that when i.i.d. symbols are drawn according to distribution $T$, type $T$ has the highest probability to be the type of the generated string. Since there are at most $(n+1)^c$ types, then $|C(T)|2^{-nH(T)} \geq \frac{1}{(n+1)^c}$ which implies the result.

To finish the proof assume that the symbols $X_1, X_2, ..., X_n$ are drawn i.i.d. from distribution $T$. Let $T_1$ be a fixed type and let $n_1, n_2, .., n_c$ be integers such that $T_1 = (\frac{n_1}{n}, \frac{n_2}{n}, ..., \frac{n_c}{n})$. Then the number of strings of type $T_1$ is equal to

$$\binom{n}{n_1, n_2, ..., n_c} = \binom{n}{n_1}\binom{n-n_1}{n_2}...\binom{n_c}{n_c}$$

Also let $T = (\frac{m_1}{n}, \frac{m_2}{n}, ..., \frac{m_c}{n})$. Then we calculate the ratio of probability of generating a string in $T$ to that of a string in $T_1$.

$$\frac{Pr(T)}{Pr(T_1)} = \frac{\binom{n}{m_1,m_2,...,m_c}\prod_{a\in\Sigma} T(a)^{nT(a)}}{\binom{n}{n_1,n_2,...,n_c}\prod_{a\in\Sigma} T(a)^{nT_1(a)}} \tag{2.11}$$

$$= \prod_{a\in\Sigma} \frac{(nT_1(a))!}{(nT(a))!} T(a)^{n(T(a)-T_1(a))} \tag{2.12}$$

$$\geq \prod_{a\in\Sigma} (nT(a))^{n(T_1(a)-T(a))} T(a)^{n(T(a)-T_1(a))} \tag{2.13}$$

$$= \prod_{a\in\Sigma} n^{n(T_1(a)-T(a))} \tag{2.14}$$

$$= n^{n\sum_{a\in\Sigma}(T_1(a)-T(a))} \tag{2.15}$$

$$= n^{n(1-1)} \tag{2.16}$$

$$= 1 \tag{2.17}$$

where inequality (2.13) follows from the inequality $\frac{a!}{b!} \geq b^{a-b}$. ■

### 2.3.1 Conditional types

In this subsection we define conditional types. Recall that type of a string can be considered a distribution on the corresponding alphabet. For strings $U \in \Sigma^n$ and $V \in \Pi^n$, conditional type of $U$ given $V$ can be considered as a set of distributions on $\Sigma$ indexed by elements of $\Pi$. To define conditional types formally first we need to define joint types.

For strings $U \in \Sigma^n$ and $V \in \Pi^n$ define string $U \times V \in (\Sigma \times \Pi)^n$ as a string of length $n$ such that $(U \times V)_j = (U_j, V_j)$. Then the joint type of $U$ and $V$ is defined as the type of the string $U \times V$ .

A string $U \in \Sigma^n$ has *conditional type* $R$ given $V \in \Pi^n$ if for any $a \in \Sigma$ and $b \in \Pi$,

$$N(a, b|U, V) = N(b|V)R(a|b).$$

If for some $b \in \Pi$, $N(b|V) = 0$, then $R$ is not uniquely determined but the set of all $U \in \Sigma^n$ having conditional type $R$, given $V$, is unique.

From this definition it is clear that for each $b \in \Pi$, $R(\cdot|b)$ is a distribution on $\Sigma$. So $R$ can be represented by a matrix, rows and columns of which are respectively indexed by $\Sigma$ and $\Pi$. Since the total value of the entries of each column of this matrix is one, this will be a *stochastic matrix*. This alternative representation of a conditional type is useful.

The set of all strings $U$ that have conditional type $R$ given $V$ is called the $R$-shell of $V$ and denoted by $C_R(V)$. Corresponding to theorems 4, 5, we have the following results on conditional types. The proofs are omitted here but they are similar to the proofs of theorems 4, 5.

**Theorem 6** *Let $V \in \Pi^n$ and $X_1, X_2, ..., X_n$ be chosen independently from an alphabet $\Sigma$ according to a stochastic matrix $\mathcal{W}(\cdot|\cdot)$ that gives a distribution on $\Sigma$ conditioned on string $V$. Then the probability of $U = X_1X_2...X_n$ depends only on $T_V$, and $T_{(U,V)}$ and is equal to*

$$2^{-n(H(T_U|T_V)+D(T_U||\mathcal{W}|T_V))}$$

*$T_V$ denotes type of $V$ and $T_{(U,V)}$ is the joint type of $U$ and $V$. Given $T_V$ and $T_{(U,V)}$, $T_U$ is uniquely determined.*

**Theorem 7**

$$\frac{2^{nH(T_U|T_V)}}{(n+1)^{-|\Sigma||\Pi|}} \leq |C_R(V)| \leq 2^{nH(T_U|T_V)}.$$

## 2.4 Game Theory

Game theory is a branch of mathematics with applications in many different disciplines from social sciences to engineering. Game theory tries to understand the behavior of systems in which different players (e.g. humans, computers, etc.) work toward maximizing their own benefit. Game theory required for our purpose is limited to a basic and well-studied subarea of this field known as *two person zero sum games*. This tool has been used frequently in information theory and information hiding. In this section we start a quick review of game

theoretic tools and concepts that are needed in the rest of this thesis. For a more in depth discussion the reader may refer to [38].

In a two person zero sum game, there are two *players* playing a game with each other. In the case that we are interested in, this game has one round which means each player makes one choice and then the result of the game can be computed. Each player has a set of *strategies* and chooses one of his strategies at the same time as the other player, with no information about the other player's chosen strategy. Formally a game is a tuple $(X, Y, P)$ where $X$ is the set of strategies of the *first player*. $Y$ is the set of strategies of the *second player* and $P : X \times Y \to \Re$ is a function called *payoff* of the game. We assume that when players choose their strategies $x \in X, y \in Y$, the second player pays $P(x, y)$ to the first player. So the gain of the first player is equal to the loss of the second player and that's the reason for the term *zero sum*.

The games with finite $X$ and $Y$ are called *finite* or *matrix games*. The reason for the second term is that such a game can be represented by a matrix, the rows and columns of which are indexed by elements of $X$ and $Y$ and the element in row $x$ and column $y$ is $P(x, y)$. Such a representation is called *the matrix representation of the game*.

The discussion below is precise for matrix games. For infinite games there is a subtlety that we ignore during the discussion but we point it out here. In a finite game values such as $\max_x \min_y P(x, y)$ do exist. For an infinite game such values may not be reachable but any value arbitrarily close to $\sup_x \inf_y P(x, y)$ is reachable. For the sake of brevity, we ignore this point throughout the following discussion.

From the point of view of the first player, when he chooses strategy $x_1$, his guaranteed gain is $\min_{y \in Y} P(x_1, y)$. So he can guarantee a gain of $\max_{x \in X} \min_{y \in Y} P(x, y)$ by choosing the right strategy. The strategy $\hat{x}$ that achieves the maximum in $\max_{x \in X} \min_{y \in Y} P(x, y)$ is called the *maxmin strategy* of first player. Similarly for the second player one can define a *minmax strategy*.

If it happens that

$$\max_{x \in X} \min_{y \in Y} P(x, y) = \min_{y \in Y} \max_{x \in X} P(x, y)$$

then this value is called *the value of the game* and is denoted by $v$. Also there exists strategies $\hat{x}, \hat{y}$ such that $v = P(\hat{x}, \hat{y})$. The pair $(\hat{x}, \hat{y})$ is called a *saddle point* of the game. It is easy to construct games with no saddle point. For example assume $X = Y = \{1, 2\}$ and $P(1, 1) = 1, P(1, 2) = 0, P(2, 1) = 0, P(2, 2) = 1$.

When a player chooses a fixed strategy it is said that he plays a *pure strategy*. Alternatively a player can play a *mixed strategy* by choosing a distribution over all available strategies. Then the expected payoff of the game will be $E[P(x, y)]$ in which the expectation is over random choices of $x$ and $y$ according to players' mixed strategies.

When matrix games are extended with the notion of mixed strategies the existence of saddle point for them is guaranteed. In this case, for the first player when playing mixed strategy $x^*$, the guaranteed gain is equal to $\min_{y \in Y} E_{x \in x^*}[P(x, y)]$, where $x$ is drawn from $x^*$ randomly. This quantity is equal to $\min_{y^*} E_{x \in x^*, y \in y^*}[P(x, y)]$. Consequently the first player can guarantee a minimum gain of $\max_{x^*} \min_{y \in Y} E_{x \in x^*}[P(x, y)]$. Similarly the second player can guarantee a maximum loss of $\min_{y^*} \max_x E_{y \in y^*}[P(x, y)]$. For all matrix games we have

$$\max_{x^*} \min_{y \in Y} E_{x \in x^*}[P(x, y)] = \max_{x^*} \min_{y^*} E_{x \in x^*, y \in Y^*}[P(x, y)] = \min_{y^*} \max_{x} E_{y \in y^*}[P(x, y)].$$

This value is called the value of the game as before. Also $(x^*, y^*)$ the maximizing and minimizing strategies for the two pirates, is called a mixed saddle point of the game. In 1928 von Neumann [35] proved that any matrix game has a mixed saddle point. The original proof by von Neumann was "a long and difficult existence proof, based on functional calculus and topology, of the 'solution' for all two-person, zero-sum, games with a finite number of strategies"[32].

This theorem is the basis of some of our experimental results in chapter 3 so we present the proof here. The proof presented here, adopted from [38] is a relatively simple proof based on linear programming and it is a constructive proof that finds the mixed saddle point. We are not aware who came up with this proof first.

### 2.4.1 Minimax theorem

**Theorem 8** *Any matrix game has a mixed saddle point.*

**Proof:** Consider a matrix game $(X, Y, P)$ and assume $A$ is the matrix representation of the game. For now assume that all entries in $A$ are positive. Consider the following linear programming problem (2.18) and its dual (2.19)

$$minimize\ xu,\ xA \geq w,\ x \geq 0 \tag{2.18}$$

$$maximize\ yw,\ Ay \leq u,\ y \geq 0 \tag{2.19}$$

Here $u = (1, 1, ..., 1) \in \Re^{|X|}$ and $w = (1, 1, ..., 1) \in \Re^{|Y|}$. The assumption that all entries of $A$ are positive, trivially implies that (2.18) has a feasible solution. Also all zero vector is a feasible solution for (2.19). Feasibility of (2.18), (2.19) implies that both problems have optimal solutions $\hat{x}$, $\hat{y}$ and $\hat{x}u = \hat{y}w = \theta$. Now we claim that $(x^*, y^*)$ where $x^* = \hat{x}/\theta \;\; y^* = \hat{y}/\theta$, is a mixed saddle point for game $(X, Y, P)$. It is easy to check that $x^*$ and $y^*$ are legitimate mixed strategies for players of the game. Assume $x$ and $y$ are arbitrary mixed strategies for the first and second player. Using (2.18), (2.19)

$$P(x^*, y) = (x^*A)y = \frac{\hat{x}Ay}{\theta} \geq \frac{wy}{\theta} = \frac{1}{\theta} \tag{2.20}$$

$$P(x, y^*) = x(Ay^*) = \frac{xA\hat{y}}{\theta} \leq \frac{xu}{\theta} = \frac{1}{\theta} \tag{2.21}$$

In particular when we choose $x = x^*$ and $y = y^*$ both sides of (2.20), (2.21) are equal, so inequalities in (2.20), (2.21) must be equal in that case. This observation means $P(x^*, y^*) = \frac{1}{\theta}$ and (2.21), (2.21) imply that $(x^*, y^*)$ is a mixed saddle point of the game.

Now if $A$ has a non-positive entry, choose $c$ such that $P(x, y) + c$ is positive for all $x, y$. Define a new game $(X, Y, Q)$ such that $Q(x, y) = P(x, y) + c$. It is not hard to see that the new game has a saddle point if and only if the original game has a saddle point and our proof is applicable to the new game. ■

### 2.4.2 Infinite games

In the game $(X, Y, P)$, if sets $X$ and $Y$ are infinite the game is called *infinite*. We are interested in the case that $X, Y \subseteq \Re^n$ and $P$ is a continuous function on $X \times Y$. Infinite games with a continuous payoff function admit a saddle point. We state this result here without a proof. The reader may find a proof in chapter two of [38].

**Theorem 9** *Any infinite game $(X, Y, P)$ where $X$ and $Y$ are metric compact sets and $H$ is a continuous function on $X \times Y$ has a mixed saddle point.*

Of particular interest to us are so called *convex games*. These are infinite games with a continuous payoff function which is convex in $y$ for any fixed $x \in X$. All convex games have saddle points in which the second player has a pure strategy. As we will see, this result yields an alternative proof for one of our theorems in chapter 4, but we omit the proof here. Another property of convex games that is interesting for us is the following.

**Theorem 10** *In the convex game* $(X, Y, P)$, $Y \subseteq \Re^n$ *the first player has an optimal mixed strategy with support of size not more than* $n + 1$.

The proof is omitted here.

# Chapter 3

# Context

While in cryptography we are interested in hiding the *content* of a message being transmitted, there are applications in which we are interested in hiding the very existence of a message. As an example consider a spy who works in a hostile country. To maintain her safety, the spy prefers to hide any information that she exchanges with her country. In this case a ciphered message can only worsen the situation and arouse more suspicion. In some other applications, the hidden message should also have some properties. For example consider a simple system of copyright protection for pictures that embeds a serial number in each copy of the picture. In this case, the message should be hidden, i.e. it should be imperceptible. Also, it should be long enough so that no two copies of the picture have the same serial number. The fact that each two copies of the picture have different serial numbers suggests that embedding such a serial number may have more complications that embedding a unique message in a picture or any other media.

This kind of problems are not new and since ancient ages people have been looking for techniques for information hiding. In 440 B.C. Histiaeus, the tyrant of Miletus, an ancient city in what is now Aydin Province of Turkey, tattooed a message on the shaved head of a trusted slave. After the hair had grown, the message was hidden. The technique has been used as recently as 20th century by some German spies. Look at [37] and references therein.

According to [5] in 1980's Margaret Thatcher ordered word processors be programmed to embed a unique key in the word spacing of documents, so that members of the cabinet who leak a document can be traced back.

Despite this ancient need for *information hiding*, it is only slightly more than a decade that the problem is studied seriously as an independent field of research in academia. The

first international workshop on information hiding was held in 1996 in Cambridge, UK [4]. The purpose of this workshop was to be a place for five or more research communities who worked on information hiding and were unaware of each other. Among other things, this workshop produced a standard terminology for the field of information hiding. See foreword of [4].

In this chapter we look at some of the research problems studied in the area of information hiding. We mostly focus on fingerprinting under marking assumption which is the subject of this thesis.

First we take a quick look at traitor tracing problem, which has a more cryptographic nature. Historically traitor tracing is the father of fingerprinting, and some other problems studied in the field of information hiding. Then we took a close look at fingerprinting. We see formal definition and an overview of the results on fingerprinting. A sibling of fingerprinting codes is the family of codes with Identifiable Parent Property, also known as, IPP codes. They are both outcomes of research on traitor tracing problem. We will quickly look at some problems studied in this in this area as well. At the end we take a look at steganography. This is a more or less different problem in information hiding that is a complementary problem for fingerprinting. As we will see soon, in the research on fingerprinting the problem of how a fingerprint is embedded in a document is not addressed. This is exactly the problem studied in steganography.

## 3.1 Traitor Tracing

Traitor tracing was first introduced in [16]. This is the paper that through time originated various, now independent, research problems, one of them being fingerprinting under marking assumption. The purpose of traitor tracing is copyright protection. Indeed earlier traitor tracing schemes can be viewed as a fingerprinting scheme under marking assumption over large alphabets, where the accusation algorithm is Hamming distance[1]. This observation has led to construction of traitor tracing schemes based on fingerprinting schemes. For example Boneh and Naor [11] used fingerprinting code constructions of Boneh and Shaw [12] and Tardos [44] to construct their traitor tracing scheme.

A traitor tracing scheme is supposed to securely distribute a *plain text* among a number

[1]Some authors restrict the definition of a traitor tracing scheme to one with Hamming distance as tracing algorithm. For example see [40].

of *authorized users*. Here we review one of the constructions of [16] to see how a traitor tracing system works. The purpose of the construction is to distribute a plain text among a set of at most $n$ users. The plain text is partitioned to $m = \log n$ segments. The *distributor* chooses two keys for each segment. Each user receives a *decoder*. Each decoder has exactly one of the two keys of each segment. The distributor encrypts each segment using each of the keys and broadcasts all the $2m$ *encrypted segments*. Each users receives all $2m$ encrypted segments and use the decoder to discard $m$ of them and decode the other $m$ encrypted segments. If we identify each of the two keys of a fixed segment using $\{0, 1\}$, then the set of $m$ keys of a user is corresponding to a binary number with $\log n$ digits. So for $n$ users the distriutor can make $n$ distinct decoders using different keys. Now If a user makes a new decoder based on her own decoder and give it to an unauthorized user, the decoder can be traced back to the person who built it using the set of keys used in the decoder.

This scheme can be generalized to be secure against coalition attacks by at most $k$ users. In this kind of attack, $k$ user get together and build a new decoder using possible all the decoders at their disposal. The distributor chooses $r = 2mk^2$ keys in advance and each user receives a subset of size $m$ of keys. To distribute keys, the set of all keys is partitioned to $m$ subsets $K_1 \cdots K_m$ of size $2k^2$ and each user receives one key from each subset which is chosen uniformly and independently at random.

Assume that the distributor wants to send out one segment of plain text to authorized users. This segment is encrypted to the *ciphertext* using a key $s$. This key is then broken down to $m$ keys $s_1 \cdots s_m$ such that $s$ is bitwise XOR of $s_1, \cdots, s_m$. Each key $s_i$ is encrypted by all keys in the set $K_i$. All these encrypted keys ($2mk^2$ encrypted keys in total) is sent to each user along with the ciphertext. Now an authorized user can decrypt $s_i$ using their keys, construct $s$ and decrypt the plaintext.

A *pirate decoder* is an unauthorized decoder capable of decrypting the ciphertext. We show if a pirate decoder is discovered one of the following two alternatives must hold.

1. Pirate decoder has access to all keys required to decode the ciphertext. This is at least one key from each $K_i$.

2. Pirate decoder needs no key to decode the ciphertext, i.e. it breaks the underlying encryption scheme.

Assuming the underlying encryption scheme is hard to break, we find out the $m$ keys that

pirate decoder uses to decode the ciphertext[2]. Then assuming there were no more than $k$ authorized users who leaked their keys, any authorized user who has more than $m/k$ common keys with the pirate decoder is accused of leaking their keys.

This scheme can be considered as a fingerprinting scheme which is secure against coalitions of size at most $k$ and has an alphabet of size $2k^2$.

Given the correspondence of this construction with fingerprinting over large alphabet, it is possible to improve this construction by using better fingerprinting schemes. In particular, since $k$-fingerprinting is possible with binary alphabet, number of keys in this construction can be reduced to $2m$.

## 3.2 Fingerprinting

There are at least three different types of *fingerprinting*. The first and the most familiar type is human fingerprinting which needs no explanation. The second type of fingerprinting has a meaning similar to *hashing*. As an example there is a technology for audio search based on fingerprinting (of the second type). This technology allows a user who hears a music (in a restaurant, a shopping mall, etc.) to find the album and the artist by calling a number on their cell phone and keeping the connection for a short period of time, say fifteen seconds [47]. In the call center, a fingerprint of the received piece of music is generated and is compared against fingerprints of music records in the database. If a match is found the artist and album of the match is sent to the user by text message.

Another application of this type of fingerprinting is in copyright protection. For this purpose, a software monitors a media stream and detects music played and compares its fingerprint against the database.

There are interesting technical problems in generating this type of fingerprint, but this type of fingerprint is different from the third type that we have considered in this thesis. The purpose of this third type of fingerprint is protection of copyright or secrecy of a document. Even though there is some overlap in the applications of the second and third kind of fingerprint, namely copyright protection, they are completely different.

In order to protect their copyright, content producers may like to make each copy of their productions unique. This would allow them to trace each copy to its original customer

[2]This is not a trivial task, but we skip details here.

(user). A natural way of implementing this idea is to embed a serial number in the product. Of course, a user must not be able to detect the serial number, otherwise he can randomly change it and make an illegal copy that can not be traced back to the malicious user i.e. *pirate*.

Even if we have embedded serial numbers that cannot be detected by a single user, coalitions of users might be able to beat the system. A group of pirates can compare their copies. Any difference that they detect in similar positions of their copies may be an indication of the embedded serial number. They can produce an illegal copy by changing the serial number in these detected positions according to a strategy of their choice. This is called a *collusion attack*. The specific strategy that pirates use to produce the illegal copy is called *pirate strategy*.

Now the content producer cannot simply check serial numbers and find the culprit. Despite this, there is some hope for the content producer to catch the pirates. There are certain positions in the serial numbers where all members of a group of pirates see similar symbols. These positions of the serial number remain undetected by pirates. So they cannot produce an illegal copy with a completely random serial number. The content producer hopes to use the undetected portion of serial number to catch the pirates. For this to work, the "serial numbers" (i.e., codewords) have to be carefully selected. Such a set of codewords (in fact, a randomized strategy to obtain them) and an *accusation algorithm* to catch at least one of the pirates is called a *fingerprinting code* if it is secure against a limited number of pirates. The mathematical definition (see below) was first given by Boneh and Shaw [12].

### 3.2.1 The model

We start with introducing simple notation. We fix a finite alphabet $\Sigma$. In this thesis we consider the *binary alphabet* $\Sigma = \{0, 1\}$ but the definitions below make sense for any alphabet and fingerprinting is studied over larger alphabets too. For a positive integer $n$ we denote the set $\{1, \ldots, n\}$ of positive integers not exceeding $n$ by $[n]$. For a sequence $X$ of length $n$ and $i \in [n]$ we denote the $i$-th entry in $X$ by $X_i$, i.e., $X = (X_1, \ldots, X_n)$.

**Definitions and Notation:** A *fingerprinting code* of length $n$ over the alphabet $\Sigma$ for the users $1, \cdots, N$ consists of two components. A *code generation algorithm* and an *accusation algorithm*.

Code generation algorithm generates codewords $x_v \in \Sigma^n$ for users $v \in [N]$ and possibly

some additional input for the accusation algorithm. Before defining the second component we need some extra definitions and notation. $R = \frac{\log_{|\Sigma|} N}{n}$ is called the *rate* of a fingerprinting code.

Codeword for the user $i \in [N]$ is denoted by $x_i$. Individual bits of this codeword would be denoted by $x_{ij}$ for $1 \leq j \leq n$. Correspondingly, random variables $X_i$ and $X_{ij}$ denote the codeword for the user $i$ and the $j$-th bit of the codeword for the user $i$.

A subset of users may use a *pirate strategy* to generate a *forged copy.* A *pirate-strategy* for a set $T$ of *pirates* and for codes of length $n$ over an alphabet $\Sigma$ is any (deterministic or randomized) algorithm that takes as input the codewords $x_v \in \Sigma^n$ of the pirates $v \in T$ and outputs a *forged codeword* $f \in \Sigma^n$ that satisfies the so-called *marking assumption*, i.e., if for some $j \in [n]$ the digits $x_{vj}$ agree for all $v \in T$, then we also have $f_j = x_{vj}$.

The second component of fingerprinting code, the accusation algorithm, takes the output of *code generation algorithm* and a *forged codeword* $f \in \Sigma^n$ and outputs a set $A(f) \subseteq [N]$ of *accused users.* If $f$ was obtained by the set $T \subseteq [N]$ of users performing a pirate-strategy and $v \in A(F)$ for some user $v \notin T$, then we say that $v$ is *falsely accused.* If $f$ obtained by a pirate-strategy of $T \subseteq [N]$ but $A(f) \cap T = \emptyset$, then we say the pirates *are not caught.*

We call a fingerprinting code *$\epsilon$-secure against $t$ pirates* if for any set $T \subseteq [N]$ of users of size $|T| \leq t$ using any pirate-strategy the probability that either the pirates are not caught or some user is falsely accused is at most $\epsilon$. This probability is over the random choices in the fingerprinting code generation and possibly in the pirate-strategy.

A *fingerprinting scheme* over the alphabet $\Sigma$ is an infinite sequence of fingerprinting codes $C_i$ over $\Sigma$ for the set $[N_i]$ of users such that $N_i$ goes to infinity. The *rate* of such a scheme is $\limsup R_i$, where $R_i$ is the rate of $C_i$. We say that a fingerprinting scheme is *$t$-secure* if $C_i$ is $\epsilon_i$-secure against $t$ pirates with $\lim_{i\to\infty} \epsilon_i = 0$. The $t$-fingerprinting capacity is the maximum achievable rate of $t$-secure fingerprinting schemes.

For a binary string $x$, $|x|$ denotes the number of ones in $x$. We repeatedly use the $\tilde{O}$ notation for running time of algorithms. $\tilde{O}$ suppresses a polylog factor compared to the standard big $O$ notation, i.e. $\tilde{O}(f(n)) = O(f(n)\log^k(n))$ where $k$ is a constant independent of $n$.

Some comments about this definition are necessary.

- The two type of errors that we mentioned above, naturally come up in the proofs. Usually proofs show that there exists a property A such that some pirates have property

A and no innocent user has property A.

- Any fingerprinting code construction must include randomization in some way. It can be easily seen that a deterministic fingerprinting code does not work. To see why, consider a binary[3] fingerprinting code for a set of three users. Consider the codeword $f$ such that $f_i$ is equal to $maj(x_{1i}, x_{2i}, x_{3i})$ where $maj$ denotes majority. This codeword may be generated by any coalition of size two of the above three users without violating marking assumption and accusation algorithm does not have any way to specify which user is not in the set of pirates. Randomness is necessary in the construction of fingerprinting codes in order to escape this trap.

- Pirate strategy might be randomized, but in the proofs, when necessary, we can limit ourselves to deterministic strategies. This is because a fingerprinting code that is secure against any deterministic pirate strategy is secure against any randomized pirate strategy as well. Any randomized strategy can be considered as a distribution over a set of deterministic strategies and the error probability of the randomized strategy is the expected value of error probabilities of underlying deterministic strategies.

The goal of fingerprinting research is to find efficient and secure fingerprinting codes. The paramount problem in the application of fingerprinting codes is the high cost of embedding every single digit of the code. This makes it important to design secure fingerprinting codes that are short, or equivalently, have high rate. In particular, recent research focused on finding or estimating the $t$-fingerprinting capacity for various values of $t$.

### 3.2.2 Lower bounds on fingerprinting capacity

Earlier results on fingerprinting under marking assumption did not use the terminology that we use here. These works were based on probabilistic analysis techniques and results were stated in terms of lower bounds and upper bounds on codeword length. Recent results on fingerprinting has used information theoretic techniques widely so the results are stated in terms of information theoretic concepts like *rate* and *capacity*.

Boneh and Shaw introduced fingerprinting under marking assumption in [12], based on results of [16] on traitor tracing. The code scheme constructed in [12] is of length

[3]The code does not need to be binary for the argument to work. We consider binary codes because our focus is on binary codes.

$O(t^4 \log N\epsilon \log \frac{2tL}{\epsilon})$ where $L = 2t \log \frac{2NL}{\epsilon}$. This can be translated to a scheme with rate $O(\frac{1}{t^4})$. The construction has two steps, first a simpler but less efficient scheme is constructed and proved to work. Then code concatenation with random error correcting codes is used to improve the parameters of the construction.

Barg, Blakley and Katabiansky [8] has used techniques from error correcting codes to improve the results of [12].

Tardos [44] introduced the notion of bias based code generation that we will see in the next chapter. This technique simplified code generation considerably as there is no need to construct fingerprinting schemes based on error correcting codes. The constructions of [44], results in codes of length $100t^2 \log(N/\epsilon)$ which corresponds to rate $\frac{1}{100t^2}$. This is the first construction that achieves the optimal order of magnitude but constant factor optimization is still possible.

The constant 100 in the length $100t^2 \log(N/\epsilon)$ was subsequently improved by several papers. In particular we are aware of [43], [42], and [10]. All these papers go through the proof in [44] and optimize various parameters in the proof to improve the codelength without fundamentally changing the code construction or the accusation algorithm. All improvements in the codelength directly translates to the same improvement in the rate of $t$-secure schemes. The biggest improvement factor, almost 10, was achieved by Skoric, Katzenbeisser and Celik in [42], but they make a reasonable simplifying assumption without a mathematical justification. The smaller improvement factors of Skoric, Vladimirova, Celik and Talstra in [43] and Blayer and Tassa in [10] is based largely on experimental evidence, although the latter paper rigorously extracts some formulas for the parameters in the proof of [44], but in the last step they also refuge to experiment to estimate their improvement factor: it is between 4 and 5. All these improvement factors state the improvement over Tardos' original result [44].

In a different approach to the problem people have tried to find a high rate construction for $t$-fingerprinting for small $t$. Here the hope is that the techniques will be then generalizable to arbitrary $t$. An early paper in this direction is [41], which achieved rate 0.026 for two pirates, using $(2, 2)$-separating codes.

The rate of 2-fingerprinting was improved to 0.2075 [9] by using randomly generated codes. Based on this result, Anthapadmanabhan, Barg [6] constructed a $t$-fingerprinting scheme which, in particular, achieved the best known rate for three pirates at the time: 0.064. Continuing this line of work, Anthapadmanabhan, Barg and Dumer [7] constructed

a $t$-secure fingerprinting schemes whose rates for $t = 2$ and 3 are much higher than previously obtained rates but the rate of their schemes deteriorates exponentially with $t$. They use independent uniform random codewords for two and three pirates, but they use the notion of *typicality* much more extensively in their analysis that allows them to achieve rates 0.25 and 0.083 for 2 and 3-fingerprinting.

Dumer [21] Considered a simplified version of fingerprinting under marking assumption in which all codewords are chosen uniformly at random from among strings with Hamming weight $p$. In this model, pirates are restricted to generating forged copies of Hamming weight $p$ as well. The scheme in [21] has a rate $\geq \frac{0.69}{t^2}$. The code construction of this scheme is similar to bias based code construction with single bias $p$. As it has been shown in [3], achieving capacity using bias based code generation, requires using no less than $O(\sqrt{\frac{t}{4\ln(2)\log(t)}})$ different bias values when pirates are allowed to use any strategy that respects marking assumption.

Another attempt to improve the result of [7] was done by Tardos and the author [3]. We used the bias based code generation technique of [44] and the notion of typicality that was previously used in [7] to achieve the best known rate for fingerprinting. We conjecture that our code achieves the optimal rate (i.e. capacity) for $t$-fingerprinting for any $t$. (For $t = 2$ the same rate was already achieved in [7]). Independently of this work [27] based on earlier construction in [34] constructed a similar scheme which has similar code construction as [3]. The construction of [3] will be presented in the next chapter of this thesis.

The main drawback of constructions of [3, 27] is the slow accusation algorithm, which is exponential in $t$. As a result of this weakness there has been attempts to improve the rate of the construction in [44] and subsequent papers [42, 43, 10], and keep the running time linear. Huang and Moulin [26], building on [34], construct a family of suboptimal fingerprinting scheme that has a linear time accusation algorithm. The rate of their construction is higher than that of [44] and the construction has game-theoretic nature like those of [3, 27].

We improve this result and unify the constructions of [3, 27, 26] by constructing a family of fingerprinting schemes parameterized by $k \in \{1, 2, ..., t\}$. While the larger $k$ results in a code with higher rate, running time is also higher namely $\tilde{O}(N^k)$. Also in the case of two pirates and $k = 1$ we improve the accusation algorithm to achieve the optimal rate in linear time. All previous constructions that achieve optimal rate for 2-fingerprinting have accusation algorithms of $\tilde{O}(N^2)$ time complexity.

The accusation algorithm of our 2-fingerprinting code has two steps. In the first step

user's codewords are sorted by Hamming distance to the forged copy. If there is user whose codeword is close enough to the forged copy, the user is accused as pirate, otherwise we run the algorithm of [3] on selected pairs of users.

Even though minimum Hamming distance sounds to be the simplest and most natural accusation algorithm it is not yet analyzed in the case of binary alphabets. We are aware of only one study of minimum Hamming distance as accusation algorithm in [33] that considers the special case of two and three pirates and uniformly random fingerprints.

In constructions of traitor tracing codes collusion-secure fingerprinting codes are the central component. In earlier constructions minimum Hamming distance was the accusation algorithm used to detect pirates, so these constructions include a fingerprinting code with minimum Hamming distance as accusation algorithm. The constructions that we are aware of use alphabets of large size.

### 3.2.3 Upper bounds on fingerprinting capacity

Upper bounds on capacity can be translated to lower bounds on code length. Boneh and Shaw [12] proved the first lower bound on the length of fingerprinting codes. It is hard to translate this bound to an upper bound on capacity. Later Tardos [44] showed a stronger lower bound on the length of fingerprinting codes that achieved the optimal order of magnitude $O(t^2 \log \frac{n}{\epsilon})$, but it was not immediately translatable to an upper bound on capacity. This result was later improved in [3] to show fingerprinting capacity is of $O(1/t^2)$.

Anthapadmanabhan, Barg and Dumer [7] are the first to prove upper bounds on the capacity of fingerprinting. The upper bounds in their paper is given in terms of a hard to evaluate information theoretic min max formula. They estimate this formula and prove strong upper bounds on the $t$-fingerprinting capacity for small values of $t$ (namely 2 and 3) and an $O(1/t)$ asymptotic bound.

We conjecture that the rate achieved in [3, 27] is actually the capacity. Proving or disproving this conjecture is a major open problem in this line of research.

### 3.2.4 Codes with identifiable parent property

Previously we showed that *binary* fingerprinting codes cannot be deterministic. This proof can be generalized to fingerprinting codes over alphabets of arbitrary size. The essential point in the proof is that in a fingerprinting code a group of pirates who see different

symbols in a column can use any symbol of the alphabet in that column of the forged copy. If we restrict the set of eligible pirate strategies, then we reach a restricted version of fingerprinting codes for which deterministic codes exist. This restricted family is called codes with identifiable parent property. In this codes in any column pirates can output one of the symbols that they see in their own codewords.

This family of codes was introduced in [24]. Such codes can be compared with 2-fingerprinting codes. Let $C$ be a code of length $n$. For two codewords $a = a_1 \cdots a_n$ and $b = b_1 \cdots b_n$, Let $D(a,b) = \{c | \forall 1 \leq i \leq n, c_i \in \{a_i, b_i\}\}$. Here $D(a,b)$ is the set of *descendants* of $a$ and $b$. Also $a$ and $b$ are called parents of each member of $D(a,b)$. Then we define $C^* = \cup_{a,b \in C} D(a,b)$. We say $C$ has identifiable parent property (IPP) if for every $c \in C^*$ all pairs $(a,b)$ such that $c \in D(a,b)$ have a common member.

In [24] various results on these codes has been presented. In particular it has been shown that Reed-Solomon codes can be used to generate IPP codes. Also it has been shown that $F(n,q)$, the length of maximum $q$-ary IPP codes of length $n$, is larger than both $q^{\lceil \frac{n}{4} \rceil}$ and $0.4(\frac{q}{4})^{\frac{n}{3}}$. Also it is proved that $F(n,q) \leq 3q^{\lceil \frac{n}{3} \rceil}$ and $F(3,q) \leq 3q - 1$.

In [28] authors have determined the precise value of $F(3,q)$. More specifically let $(r+1)^2 - 1 \leq q \leq (r+2)^2 - 1$. Then we can write $q = r^2 + r + k$ where $0 \leq k \leq 2r + 2$. Then authors show that for $q \geq 24$, the size of a maximum IPP $q$-ary code of length three is equal to $3r^2 + m$ where $m$ is either zero or $3k - 6 \leq m \leq 3k - 2$. The exact value of $m$ depends on value of $r$ and $k$. The interested reader may find the exact result in [28]. The maximum size of $q$-ary IPP codes for $q \leq 48$ was previously found in [46]. Other authors have considered decoding algorithms for IPP codes and generalizing IPP codes to arbitrary number of pirates.

## 3.3 Steganography

Steganography comes from Greek roots and literally means *covered writing* [37]. In steganography a message is *embedded* in a media in such a way that its existence is hidden from a third party. The media in which the secret message is embedded is called *cover text*. Our focus here is on the use of digital media as cover text. The general idea in steganography is to find non-significant data in covertext and modify it to embed the secret message. There are some ad hoc techniques, for example on a storage device like a hard disk, information can be written on unallocated space. Furthermore a hidden partition on a hard drive can

be used for this purpose. Actually a steganographic ext2 file system based on this idea has been implemented. See [30] and reference therein. Another technique is using unused fields of networking protocols to store secret information.

Researchers in steganography are mostly interested in using image and audio files as cover text. There is an abundance of such files being exchanged on the web and it is easy to write a software to implement an embedding algorithm for audio and image. Also exchanging audio and image files between two parties arouses little suspicion.

A very simple method of embedding secret information in an image or audio file is to set the least significant bit of the pixels of the image to the bits of the secret message. For example if an image file stores RGB information of pixels, modification in least significant bit, does not result in any perceptible change in the image. This method is called Least Significant Bit(LSB) method.

Stronger steganography techniques are based on the assumption that the embedding technique is publicly known and the secrecy is maintained by a private key known to authorized users. The LSB method can be improved by using a *one time pad* i.e. a pseudo random binary key that chooses which pixels are chosen for embedding.

A more sophisticated approach stores one bit of secret message as parity of $k$ bits of cover text [5]. While in the first glance it seems that this will reduce efficiency of steganography system, it increases the efficiency.

There are two types of attacks against an steganography system. *Passive attacks*, try to detect whether or not an innocent looking message being exchanged contains any hidden information. *Active attacks* try to destroy any potentially hidden message by using the same approach, modifying non-significant parts of the message being exchanged.

The basic ideas of steganography that we mentioned above are resistent against passive attacks but they are very vulnerable when exposed to any active attack. These ideas are examples of steganographic systems that are known as *substitution system*. The upside of these ideas is that they are easy to implement. Authors of [29] enumerate several steganographic softwares based on least significant bit substitution. There are other steganographic techniques that are robust to attacks like compression, cropping or image processing. For a survey of main steganography techniques the interested reader may refer to [29].

### 3.3.1 Theoretical questions

There are various theoretical questions studied in steganography literature. The first such question is how to define security for a steganography system and how one can prove that a steganographic system is secure. Steganography follows a standard cryptography assumption that the technique used is known publicly and security is due to a secret key shared by authroized parties. In cryptography there are public key cryptography protocols that are used to share a secret key between authorized parties. An interesting theoretical questions is to construct public key steganpgrpahy systems. For discussions regarding such results and pointers to related work see [5, 25].

# Chapter 4

# Higher Rates

In this chapter we present the construction of our fingerprinting code that appeared in [3]. The result of this chapter will also be the basis of constructions presented in the next chapter.

From a theoretical point of view, our result in this section is not fully constructive. We use bias based code generation technique of [44] to generate our code matrix, but the optimal bias distribution that results in optimal code is not known theoretically. Despite this, there are two solutions that compensate the lack of a fully constructive code generation.

1. Our construction works for any arbitrary discrete bias distribution. Even the condition of discreteness of bias distribution (that is needed for one of the proofs to work) is a technical requirement and can be removed. Using distributions other than optimal distribution results in lower rate. This rate is given in the remaining of this chapter and is in the form of minimum of a mutual information.

2. Optimal bias distribution can be approximated using numerical computation techniques. Using distributions close to the optimal distribution results in rates close to optimal. This is a consequence of continuity of mutual information as we will see in the rest of this chapter.

We solve the game numerically for small $t$ to study the high rate fingerprinting scheme for small number of pirates.

## 4.1 Basics

In this section we introduce two basic concepts that has been used in this chapter.

### 4.1.1 Bias Based Code Generation

Consider the set of users $[N]$, a distribution $D$ on $[0,1]$, and assume that we want to generate a binary codeword $x_v \in \{0,1\}^n$ for each user $v \in [N]$. This is done in a two phase process. First we pick a *bias-vector* $\bar{p} = (p_1, \ldots, p_n)$ choosing the biases $p_j \in [0,1]$ independently for $j \in [n]$, each according to distribution $D$. Then, in the second phase we pick the digits $x_{vj} \in \{0,1\}$ of the codewords independently for $v \in [N]$ and $j \in [n]$. We pick $x_{vj}$ with expectation $p_j$. The distribution $D$ is called the *bias distribution*. Notice that the bias vector is kept hidden from the users.

### 4.1.2 Channel based pirate strategies

These are strategies with the following two properties.

- Each bit $f_i$ of the forged copy depends only on the $i$-th bit of pirates' codewords.
- Dependence of $f_i$ on $x_{ji}$ is the same for every column.

A channel based strategy is characterized by a function

$$S : \{0,1\}^t \to [0,1].$$

When pirates use a channel based strategy, for each column $i$ they produce $f_i$ randomly with expectation $S(x_{1i}, x_{2i}, ..., x_{ti})$ where $x_{ji}$ denotes the $i$-th bit of $x_j$. The terms channel based strategy and channel as well as the notation for the characterizing function are used interchangeably.

Since pirates are restricted by marking assumption we can further limit ourselves to *eligible channel based strategies.* By *eligible* we mean that $S(00...0) = 0$ and $S(11...1) = 1$. This condition ensures that marking assumption is respected. We will see later in this chapter that by restricting ourselves to channel based strategies we can simplify our proofs while maintaining generality of the results.

## 4.2 The fingerprint game

We start by defining a two person zero sum game. Assume David and Paula play the following game. David's strategy is a real number $p \in [0,1]$. Paula's strategy is an eligible channel $S : \{0,1\}^t \to [0,1]$.

Using players' strategies we define a distribution $B_{p,S}$ over all $(t+1)$-tuples $(x_1, ..., x_t, f)$ as follows. Each $x_j \in \{0,1\}$ $(j \in \{1, ..., t\})$ is chosen independently with expectation $p$ and $f \in [0,1]$ is chosen such that $Pr[f = 1|x_1...x_t] = S(x_1...x_t)$.

After two players chose their strategies, Paula pays $I(x_1, ..., x_t; f)$ to David. We denote this value by $I_{p,S}$. Notice that the mutual information is calculated in the distribution $B_{p,S}$.

David and Paula may choose to use mixed strategies $D_1$ and $D_2$ where $D_1$ is a distribution over interval $[0,1]$ and $D_2$ is a distribution over eligible channels $S : \{0,1\}^t \to [0,1]$. Then we can define $B_{D_1,D_2}$ to be the following distribution. First we choose $p \in [0,1]$ randomly according to $D_1$, then we randomly choose an eligible channel $S$ according to $D_2$. Now we choose a $(t+1)$-tuple $(x_1, ..., x_t, f)$ according to $B_{p,S}$. Now the expected payoff of the game would be

$$E_{p \in D_1, S \in D_2}[I_{p,S}] = I(x_1, ..., x_t; f|p, S), \tag{4.1}$$

where the conditional mutual information in the right hand side is calculated in $B_{D_1,D_2}$. When $D_1$ is concentrated on one value $p$, this distribution is denoted by $B_{p,D_2}$. Similarly when $D_2$ is concentrated on a single value $S$, it is denoted by $B_{D_1,S}$. It turns out that Paula is always better off choosing a pure strategy. Assume Paula uses mixed strategy $D_2$. Consider the pure strategy $\overline{S}$ where $\overline{S}(x_1...x_t) = E_{D_2}[S(x_1...x_t)]$. Then for any strategy $p$ for David,

$$E_{S \in D_2}[I_{p,S}] = I(x_1, ..., x_t; f|S) = I(x_1, ..., x_t; f, S) \geq I(x_1, ..., x_t; f) = I_{p,\overline{S}}. \tag{4.2}$$

The first equality is true by definition of expected value and conditional mutual information. Notice that both sides of the second equality are calculated in $B_{p,D_2}$. The second equality is true because $S$ and $(x_1, ..., x_t)$ are independent random variables. The inequality is true because removing one random variable from a mutual information can only reduce it. $I(x_1, ..., x_t; f)$ has the same value when computed in $B_{p,\overline{S}}$ and $B_{p,D_2}$.

It is tempting to repeat the above argument to show that the distributor is also better off using a pure strategy. This does not work for the following reason. Consider the two

person game we mentioned before. In the one round of this game Paula chooses an eligible channel that will be used to choose one bit $f$. Paula can use a mixed strategy $D$ or the corresponding pure strategy $E_{S\in D}[S]$. In either case the value of $Pr[f = 0]$ is the same. This is not the case for David. As an example the uniform distribution over $\{0.001, 0.999\}$ as mixed strategy $D$ and $E_{p\in D}[p] = 1/2$ as the corresponding pure strategy. If David uses the pure strategy then for $|(x_1, ..., x_t)| = t(\frac{1}{2} \pm \epsilon)$ with high probability for some small $\epsilon$. But if David uses the mixed strategy then $|(x_1, ..., x_t)|$ will be very close to zero or $t$ with high probability.

Von Neumann minimax theorem says that any two person zero sum game has an equilibrium. Von Neumann's original versions of this theorem is for finite games. Here we need Ville's continuous version (see [39] for the original one-dimensional version and for example [36] for the generalization suitable for us) and we have to use that the payoff function is continuous. Taking the above observation that Paula is better off choosing a pure strategy, we conclude that

$$V_t := \max_{D_1} \min_{S} E_{p\in D_1}[I_{p,S}] = \min_{S} \max_{p} I_{p,S} \tag{4.3}$$

Here $D_1$ is a distribution on $p$ and $V_t$ is the *value of the game*. The value of the game is the minimum expected amount that David will receive if he plays the optimal strategy and also the maximum expected amount that Paula will pay if she plays his optimal strategy.

Lemma 12 implies that the rate of our fingerprinting scheme, denoted by $R_t$, is directly related to $V_t$

$$R_t = \frac{V_t}{t}.$$

By $D^t$ we denote the optimal distribution $D_1$ that gives the maximum in (4.3). One of the proofs, as we will see soon, requires that $D^t$ is a discrete distribution. This is indeed true. As Theorem 16 will show, the size of the support of $D^t$ has a lower bound of $\sqrt{\frac{t}{4(\ln 2)(\log t)}}$ and an upper bound of $\lfloor \frac{t}{2} \rfloor + 1$. The exact size of the support is unknown even for small values of $t \geq 4$. For $t = 2$, the optimal distribution is concentrated on the single value $\frac{1}{2}$. For $t = 3$ the support includes two values $p$ and $1 - p$. Our numerical results show that $p$ is close to 0.26.

### 4.2.1 Details of construction

We denote our code for fingerprinting with $E_{t,R,\delta,n}$. Here, $t$ is the maximum number of pirates. $R$ is the rate of the code, $n$ is codelength and $\delta$ is a security parameter.

Codewords are generated using bias based code generation with $D = D^t$. So the optimal strategy for David is used in code generation. It seems that the type of strategy we have considered for Paula in the game is much more restricted than possible strategies that can be used by pirates. For one thing, the value of $f$ at one column may depend on values of $x_i$ at other columns. But we prove that there always exists an eligible channel that is *close enough* to the pirate strategy. Assume that we generate codewords of a fingerprinting code. Consider a t-tuple $u = (x_1...x_t)$ of pirates[1] that generate a forged codeword $f$. Let $B_u$ be the joint type of the codewords of the pirates and the forged copy $f$.

Alternatively $B_u$ can be defined by choosing a $(t+1)$-tuple $(x_{1i}, ..., x_{ti}, f_i)$ where $i \in [n]$ is chosen uniformly at random and ($x_{ji}$ denotes the $i$-th bit in the codeword of $x_j$.

**Definition:** Given a t-tuple $u$ of users and the corresponding distribution $B_u$, the perceived strategy of $u$ is the conditional type of the forged copy $f$ given codewords of users in $u$. Alternatively the perceived strategy is denoted by a channel $S$ defined as $S(b) = E[f|x_u = b]$ for all $b \in \{0,1\}^t$ with $Pr[x_u = b] > 0$, where the probability and expectation are in the distribution $B_u$. For values of $b$ with $Pr[x_u = b] = 0$ we arbitrarily define $S(b) = 0$ with the exception of $S(1^t) = 1$ (to make $S$ eligible if possible).

**Lemma 11** *Consider the codewords generated for users of a fingerprinting code $E_{t,R,\delta,n}$ and a set of at most t pirates performing a pirate-strategy to produce the forged codeword. Let $u$ be a $t$-tuple of distinct users including all the pirates. Then the perceived strategy $S$ of $u$ is an eligible channel and the probability that the total variation distance of $B_u$ and $B_{D^t,S}$ is greater than $\delta$ is exponentially small.*

We present the proof of this lemma in the next section.

**Accusation Algorithm:** To accuse pirate(s) based on a given forged copy $f$ we repeat the following for any $t$-tuple of users until at least one person is accused. Fix a $t$-tuple $u$ of users and find the corresponding perceived channel $S$. If the perceived channel is not eligible then accuse no one in $u$ (but some members of $u$ might be accused later on in the course of the algorithm). Also if the variation distance between $B_u$ and $B_{D^t,S}$ is larger than $\delta$ accuse nobody in the tuple (again some users in this tuple might be accused later). Otherwise choose a minimal nonempty $w \subseteq u$ such that $I(x_w; f) \geq |w|R_t$. Then accuse all users in $w$.

[1]Here we identify users with their codewords.

Notice that such a $w$ exists because (4.3) implies $I(x_u; f) > tR_t$. Mutual informations to be interpreted in $B_{D^t,S}$.

If we can show that it is unlikely that the accusation algorithm accuses an innocent user, then Lemma 11 implies that the accusation algorithm will catch at least one of the pirates. The next lemma shows that probability of accusing an innocent user is exponentially small. In the next lemma we don't require that the number of pirates is at most $t$ only that this number is subexponential, i.e., $2^{o(n)}$. Consequently if the number of pirates exceeds the bound $t$, even though the accusation algorithm may fail to detect any pirate, no innocent user will be accused with high probability.

**Lemma 12** *For any $t$ and $R < R_t$ and small enough $\delta > 0$ the following holds for the fingerprinting codes $E_{t,R,\delta,n}$. For an arbitrary, but subexponential size set of users (the pirates) performing any pirate strategy the probability that anybody gets falsely accused is exponentially small.*

## 4.3 Proofs

In the proofs below, when we use the term *with high probability*, we mean with probability $1 - \alpha$ where $\alpha$ is exponentially small.

**Proof of Lemma 11:** First assume that instead of $B_u$ and $B_{D_t,S}$ we want to upper bound the variation distance of $C_u$ and $C_{D^t}$. Here $C_u$ is the marginal distribution of $B_u$ on $(x_1, ..., x_t, p)$. ($f$ is discarded). similarly $C_{D^t}$ is the marginal distribution of $B_{D^t,S}$ on $(x_1, ..., x_t, p)$. Since $f$ is discarded $C_{D^t}$ does not depend on $S$. This distance is easy to bound. There are finitely many values that $p$ can take[2]. Denote the number of possible values of $p$ by $k$. So a tuple $(x_1, ..., x_t, p)$ can take $2^t k$ possible different values. Using Chernoff bound one can show that probability of each value in $C_u$ and in $C_{D^t}$ differs by at most $\frac{\delta}{2^{t+1}k}$ with high probability. This is enough to show that variation distance of $C_u$ and $C_{D^t}$ is bounded from above by $\frac{\delta}{2}$ with high probability.

Now we consider a third distribution $B^*$. Marginal distribution of $B^*$ on $(x_1, ..., x_t, p)$ is $C_u$, but like $B_{D^t,S}$, $f$ is generated using channel $S$. Variation distance between $B^*$ and $B_{D^t,S}$ is the same as the variation distance of $C_u$ and $C_{D^t}$. So variation distance of $B^*$ and

[2]Here we use the assumption that David's strategy is a discrete distribution

$B_{D^t,S}$ is at most $\frac{\delta}{2}$ with high probability. It remains to bound the variation distance of $B_u$ and $B^*$ by $\frac{\delta}{2}$. Then triangle inequality gives the result required.

To bound the distance of $B^*$ and $B_u$ we consider a four step process to generate codewords.

1. First we generate the bias vector $(P_1, ..., P_n)$ according to $D^t$.
2. Generate the codewords of pirates.
3. Choose a new bias vector $P'$ from among all permutations of the vector generated in step 1 that satisfy the following condition. Joint type of the bias vector and the pirates' codewords must not change.
4. Generate the codewords for other users using bias vector $P'$.

After step 2, codewords for pirates are fixed. In the next steps, joint type of pirates' codewords and bias vector will not change. So the distribution $C_u$, as defined earlier, is fixed. In $B^*$ generation of $f$ is done according to perceived channel $S$. So at the end of step 2, $B^*$ is fixed. This is not the case for $B_u$. It is possible that the joint type of the forged copy, the pirates' codewords and the bias vector change during the third step. So $B_u$ will be fixed only after step 3.

Now we fix a value $(b, c, p_0)$ and bound the difference of its probability in $B_u$ and $B^*$. We consider the following sets.

$$H = \{j \in [n] | (X_j^{u_1}, ..., X_j^{u_t}) = b\} \tag{4.4}$$

$$H_0 = \{j \in H | F_j = c\} \tag{4.5}$$

$$H' = \{j \in H | P_j^{'} = p_0\} \tag{4.6}$$

$$H'' = \{j \in H | P_j = p_0\} \tag{4.7}$$

After the second step $H$, $H''$ and $H_0$ are fixed but $H'$ will be fixed only after the third step. The probability of occurrence of $(b, c, p_0)$ in $B_u$ is

$$\frac{|H'' \cap H_0|}{n} \tag{4.8}$$

This can be interpreted as the size $S$ of intersection of a fixed set $H_0$ with a uniform random subset $(H')$ of a given size $(|H''|)$ of a fixed base set $(H)$. The expected value of $S$ is $\frac{|H'| \cdot |H_0|}{n|H|}$ which happens to be the probability of occurrence of $(b, c, p_0)$ in $B^*$. Using Chernoff bound,

one can show that the probability that $S$ differs a lot from its expected value is exponentially small. ■

**Proof of Lemma 12:** We start by fixing a tuple $w$ of pirate indices and assuming that $x_w$ have used a pirate strategy to produce the forged copy $f$. So at this point biases, codewords for pirates and forged copy are fixed but codewords for innocent users are not yet generated.

The accusation algorithm is performed in $B_{D^t,S}$. But we need to prove that our claim holds when dealing with $B_u$. Given the continuity of mutual information and if the variation distance between $B_{D^t,S}$ and $B_u$ is small then we have $I_u(x_w; f|p) \leq I(x_w; f|p) + \delta'$. Here $I_u$ and $I$ respectively denote mutual information calculated in $B_u$ and $B_{D^t,S}$.

Assume that accusation algorithm pronounces a tuple $x_z$ of users to be pirates. We want to prove that it is unlikely that some users in $x_z$ are innocent. On the contrary assume that $x_z$ can be partitioned to two subsets $x_{z_1}$ and $x_{z_2}$ such that users in $x_{z_1}$ are pirates and those in $x_{z_2}$ are innocent. If $x_{z_2}$ is nonempty we can write

$$I(x_z; f|p) = I(x_{z_1}, x_{z_2}; f|p) = I(x_{z_1}; f|p) + I(x_{z_2}; f, x_{z_1}|p) \tag{4.9}$$

The first equality is trivial. The second equality is true because given $p$, $x_{z_1}$ and $x_{z_2}$ are independent. Now, we know that $x_z$ is a minimal set of users such that $I(x_z; f|p) \geq |z|R_t$. So $I(x_{z_1}; f|p) \leq |z_1|R_t$. Consequently, we should have $I(x_{z_2}; f, x_{z_1}|p) \geq |z_2|R_t$.

We conclude that to prove the lemma we need to show that the probability that we assign codewords to users $x_{z_2}$ such that

$$I(x_{z_2}; f, x_{z_1}|p) \geq |z_2|R_t \tag{4.10}$$

is exponentially small.

Notice that in (4.10), $x_{z_1}$, $f$ and $p$ are fixed and codewords for users in $x_{z_2}$ are selected randomly according to bias vector $p$. The fact that this set of users has been considered by the accusation algorithm implies that, the conditional type of $x_{z_2}$ given bias vector $p$ is *typical*, i.e. close to the expected type. Theorem 7 implies that the number of such choices is within a polynomial factor of $2^{nH(z_2|p)}$.

Similarly to pass the accusation algorithm conditional type of $x_{z_2}$ given $x_{z_1}$, $f$, and $p$ must be close to expected. Also the number of such choices is within a polynomial factor of $2^{nH(x_{z_2}|p,x_{z_1},f)}$.

This implies that the probability of choosing codewords $x_{z_2}$ such that (4.10) holds is

$$2^{-nI(x_{z_2};x_{z_1},f|p)+O(\log n)} \tag{4.11}$$

$$\leq 2^{-nI(x_{z_2};f|x_{z_1},p)+O(\log n)} \tag{4.12}$$

$$\leq 2^{-n|x_{z_1}|R_t+\delta' n+O(\log n)} \tag{4.13}$$

$O(\log n)$ in the exponent suppresses the polynomial factors we encountered in the counting. The first inequality is a general property of mutual information and the second one is result of (4.10).

Now we need to take a union bound. This will make sure that different choices of $x_{z_2}$ and $x_{z_1}$ are taken into account. Let $q = |z_2|$. Then there are $2^{qRn}$ choices for $x_{z_2}$. Also given that $t$ is a constant, there are constantly many choices for $x_{z_1}$. (Notice that here the only requirement on $t$ is that it is subexponential in $n$.) So the total error probability is

$$2^{(R-R_t+\delta')qn+O(\log n)}.$$

Choosing $\delta' < R_t - R$ is enough to make sure that the claim holds.■

## 4.4 Numerical Results

As the proofs in the previous section are not constructive, we study the high rate fingerprinting code for small values of $t$ through solving the game numerically. In this study we look at the value of $I_{p,S}$ and find $V_t$ as well as minimizing $S$ and maximizing values of $p$.

There are infinitely many choices for $p$ and $S$. So instead of working on the continuous interval $[0, 1]$, we consider the discretized set $\mathcal{A} = \{0, 0.01, 0.02, ..., 1\}$. To fix a pirate strategy we need to choose $2^t - 2$ values from $\mathcal{A}$. To speed up the search, we reduce this number to $t - 1$ by restricting ourselves to symmetric channels. A symmetric channel $S$, has the following two properties.

1. For each $x, y \in \{0, 1\}^t$ where $|x| = |y|$, $S(x) = S(y)$.
2. For each $x, y$ where $|x| = t - |y|$, $S(x) = 1 - S(y)$.

We need to prove that by restricting ourselves to symmetric channels, we don't miss the optimal channel. We can further speed up our search by looking at values of $p$ in $\mathcal{A} \cap [0, 1/2]$. To justify this restriction we need to show that $D^t$ can be chosen such that for any $p$, $Pr_{D^t}(p) = Pr_{D^t}(1 - p)$.

**Lemma 13** *There exists a symmetric channel $S$ such that $max_p I_{p,S} \leq V_t$. Also $D^t$ can be chosen such that for any $p \in [0,1]$, $Pr_{D^t}(p) = Pr_{D^t}(1-p)$.*

The lemma above is actually very useful and has applications beyond solving the game numerically as we will see in the next chapter. Table 4.4 below summarizes our numerical results.

| t | Support of $D^t$ | $S^t$ | $R_t$ |
|---|---|---|---|
| 2 | 0.5 | 0, 0.5, 1 | 0.25 |
| 3 | 0.26, 0.74 | 0, 0.38, 0.62, 1 | 0.0975 |
| 4 | 0.227, 0.773 | 0, 0.27, 0.5, 0.73, 1 | 0.0545 |
| 5 | 0.18, 0.82 | 0, 0.23, 0.36, 0.64, 0.77, 1 | 0.0338 |
| 6 | 0.15, 0.85 | 0, 0.19, 0.31, 0.5, 0.69, 0.81, 1 | 0.0232 |
| 7 | 0.13, 0.50, 0.87 | 0, 0.16, 0.26 , 0.44, 0.56, 0.74, 0.84, 1 | 0.0168 |

Table 4.1: Optimal channels and distributions and the rate for $t$ up to 7. The third column lists values of $S^t(x)$ in increasing order of $|x|$. All numbers for $t \geq 3$ are numeric approximations.

### 4.4.1 Proof of Lemma 13

Assume that $S$ is an optimal pirate channel. We show how to modify $S$ and make it symmetric while preserving its optimality. To achieve the first symmetry condition we define

$$S^\sigma(x_1, x_2, ..., x_t) = S(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(t)})$$

for any permutation $\sigma$ over $\{1, 2, ..., t\}$. Then define a new channel, $\bar{S}$ such that

$$\bar{S} = E_\sigma[S^\sigma]$$

where $E_\sigma$ means expectation over uniform distribution over all permutations $\sigma$. Then

$$I_{p,\bar{S}} \leq E_\sigma[I_{p,S^\sigma}] = I_{p,S}$$

The proof of the inequality is the same as Equation 4.2. The equality is proved using the fact that $I_{p,S^\sigma}$ has a fixed value independent of $\sigma$. It is easy to verify that $\bar{S}$ satisfies the first condition of symmetry.

For the second condition, again consider an optimal channel $S$. We show how to change it to achieve the second condition of symmetry while preserving optimality. Similar to the

first part define a channel $S'$ such that

$$S'(x_1, x_2, ..., x_t) = 1 - S(1 - x_1, 1 - x_2, ..., 1 - x_t).$$

Consequently we have $I_{p,S} = I_{1-p,S'}$. Now define,

$$\bar{S}(x_1, x_2, ..., x_t) = \frac{S(x_1, x_2, ..., x_t) + S'(x_1, x_2, ..., x_t)}{2}.$$

It can be seen that $\bar{S}$ is optimal while it satisfies the second condition of symmetry.

To prove the last statement of the lemma regarding $D^t$, assume $D^t$ is not symmetric and define the distribution $D_1$ such that for all $p$,

$$Pr_{D_1}(p) = Pr_{D^t}(1 - p).$$

Now define the distribution $\bar{D}$ such that

$$Pr_{\bar{D}}(p) = \frac{Pr_{D^t}(p) + Pr_{D_1}(p)}{2}.$$

We have

$$E_{p\in\bar{D}}[I_{p,S}] = \frac{E_{p\in D_1}[I_{p,S}] + E_{p\in D^t}[I_{p,S}]}{2} = \frac{E_{p\in D^t}[I_{1-p,S}] + E_{p\in D^t}[I_{p,S}]}{2} \geq V_t.$$

This proves that $\bar{D}$ is also an optimal distribution and it is symmetric.

## 4.5 Estimating $R_t$

Even though we have a formula that give us the exact value of $R_t$, the current form of this value is not informative. In this section we try to find bounds on $R_t$. We have a couple of different formulas that give us the value of $R_t$.

$$R_t = \frac{1}{t} E_{p\in D^t}[I_{p,S^t}] = \max_p I_{p,S^t} = \min_S E_{p\in D^t}[I_{p,S}] = \frac{V_t}{t}.$$

The main difficulty here is the presence of mutual information in the above formulas. As we saw in Section 2.2, mutual information can be stated as relative entropy of two probability distributions. Pinsker's inequality of section 2.2 gives us a lower bound on the value of this relative entropy. We use this tool to prove a lower bound on $R_t$.

For a fixed $p$, $I_{p,S^t}$ can be written as $D(\mathcal{P}(x_1, ..., x_t, f)||\mathcal{P}(x_1, ..., x_t)\mathcal{P}(f))$. Here $\mathcal{P}(X)$ is the distribution of random variable $X$. Remember that $x_1, ..., x_t$ are binary random variables with expectation $p$ and $f$ is a binary variable with expectation $S^t(x_1, ..., x_t)$.

Let $x$ denote the $t$-tuple $(x_1, ..., x_t)$ and define $h(c||d) = c \log \frac{c}{d} + (1-c) \log \frac{1-c}{1-d}$. Also let $g(p) = Pr(f = 1)$. Then $D(\mathcal{P}(x_1, ..., x_t, f)||\mathcal{P}(x_1, ..., x_t)\mathcal{P}(f))$ can be written as

$$\sum_{x \in \{0,1\}^t} Pr(x) h(S^t(x)||g(p)).$$

This in turn is equal to $E_p[h(S^t(x)||g(p)]$, where $E_p[\cdot]$ denotes expectation in the distribution $B_{p,S^t}$. So we have

$$I_{p,S^t} = E_p[h(S^t(x)||g(p))] \tag{4.14}$$

Now we can apply Pinsker's inequality and write

$$I_{p,S^t} \geq \frac{2}{\ln 2} E_p[(S^t(x) - g(p))^2] \tag{4.15}$$

Since we have

$$g(p) = \sum_{x \in \{0,1\}^t} S^t(x) p^{|x|} (1-p)^{t-|x|} = E_p[S^t(x)],$$

the expectation in Equation (4.15) is exactly variance of $S^t(x)$. Unfortunately calculating the variance of an unknown function of a random variable is not an easy task so we need to find a way to bound this value from below. The observation below is the solution to our problem. Let $\Delta = |x| - pt$. Then

$$E_p[S^t(x)\Delta] = (p - p^2) g'(p). \tag{4.16}$$

Remember that $g(p)$ is an unknown value but we know that $g(0) = 0$ and $g(1) = 1$ due to marking assumption. So we know that $\int_0^1 g'(p) dp = 1$. As we will see later this helps us to get around the problem of $S^t$ being unknown. Now we can use Cauchy-Schwarz inequality

$$E_p[(S^t(x) - g(p))^2] E_p[\Delta^2] \geq E_p[(S^t(x) - g(p))\Delta]^2. \tag{4.17}$$

Given that $E_p[\Delta] = 0$, we also have

$$E_p[(S^t(x) - g(p))\Delta] = E_p[S^t(x)\Delta] - g(p) E_p[\Delta] = (p - p^2) g'(p). \tag{4.18}$$

Also we have $E_p[\Delta^2] = t(p - p^2)$. This is because $E_p[|x|] = pt$, hence $E_p[\Delta^2] = E_p[(|x| - pt)^2] = Var[|x|] = Var[x_1 + x_2 + ... + x_t]$. Given $p$, variables $x_i$ are independent and each of them has variance $p - p^2$. So $E_p[\Delta^2] = t(p - p^2)$. Putting this last observation and (4.15) - (4.18) together, we conclude that

$$I_{p,S^t} \geq \frac{2}{\ln 2} \frac{E[(S^t(x) - g(p))\Delta]^2}{E[\Delta^2]} = \frac{2}{\ln 2} \frac{(p - p^2)(g'(p))^2}{t}$$

So

$$g'(p) \leq \sqrt{\frac{\ln(2)tI_{p,S^t}}{2(p-p^2)}} \leq \sqrt{\frac{\ln(2)t^2R_t}{2(p-p^2)}}. \tag{4.19}$$

Earlier we said that $\int_0^1 g'(p)dp = 1$. Using the fact that $\int_0^1 (p-p^2)^{-1/2}dp = \pi$ and taking the integral of both sides of the inequality (4.19) from zero to one we have

$$t^2R_t \geq \frac{2}{\ln(2)\pi^2}.$$

So we have proved the following theorem.

**Theorem 14**

$$R_t \geq \frac{2}{\ln 2\pi^2 t^2} = \frac{0.292...}{t^2}$$

This technique can be stretched to result in a stronger bound even though the stronger bound is an asymptotic one, unlike Theorem 14 that holds for any $t \geq 2$.

**Theorem 15**

$$R_t \geq \frac{1}{2\ln(2)t^2} + o(t^{-2}) = \frac{0.721... + o(1)}{t^2}$$

**Proof:** We use the same notation and some of the formula in the proof of Theorem 14. Recall (4.18) where we applied Cauchy-Scwarz inequality. Let $\hat{\Delta} = S^t(x) - g(p)$. We want to bound $E[\Delta\hat{\Delta}]$. This time our plan is this. We consider three different cases, depending on the value of $\Delta$ and $\hat{\Delta}$. In each case we come up with a bound for $E[\Delta\hat{\Delta}]$. Then we put these bounds together and construct a new upper bound for $g'(p)$, as we did the same thing in (4.19). After this step, our proof will be totally different from the previous proof.

Let $T_1$ and $T_2$, be two positive thresholds. We do not fix these thresholds rather we will put some constraint on their values. Then consider three different events. Event $A_1$ in which $\hat{\Delta} \leq A_1$. Event $A_2$, that $\hat{\Delta} > A_1$, but $\Delta \leq T_2$, and event $A_3$ that $\hat{\Delta} > A_1$ and $\Delta > A_2$. Let $\chi_a$ be the indicator random variable corresponding to $A_a$. Then

$$E_p[\hat{\Delta}\Delta] = \sum_{a=1}^{3} E_p[\chi_a\hat{\Delta}\Delta] \tag{4.20}$$

Now we calculate an upper bound for each term of the summation above separately. The case of $A_1$ is more complicated. First assume that $g(p) \leq 1/2$ . Now we apply the stronger form of Pinsker's inequality (Lemma 2). For $S^t(x) \leq 1/2$, if $S^t(x) \leq g(p)$,

$$h(S^t(x)||g(p)) \geq \frac{\hat{\Delta}^2}{2\ln(2)(g(p)-(g(p))^2)}.$$

Alternatively if $g(p) > S^t(x)$ and $S^t(x)$

$$h(S^t(x)||g(p)) \geq \frac{\hat{\Delta}^2}{2\ln(2)(S^t(x) - (S^t(x))^2)}.$$

Finally if $S^t(x) > 1/2$, the original form of Pinsker's inequality implies that,

$$h(S^t(x)||g(p)) \geq \frac{2\hat{\Delta}^2}{\ln 2}.$$

Assuming $A_1$ holds(i.e. $\Delta_1 \leq T_1$) we can merge all three cases to

$$h(S^t(x)||g(p)) \geq \frac{\hat{\Delta}^2}{2\ln(2)\left((g(p) - (g(p))^2) + T_1\right)}.$$

From this we conclude

$$I_{p,S^t} = E_p[h(S^t(x)||g(p))] \geq E_p[\frac{\chi_1\hat{\Delta}^2}{2\ln(2)(g(p) - g(p)^2) + T_1}] \tag{4.21}$$

Now we have

$$E_p(\chi_1\hat{\Delta}\Delta)^2 \leq E_p[\chi_1\hat{\Delta}^2]E_p[\Delta^2] = (p - p^2)tE[\chi_1\hat{\Delta}^2].$$

Combining the last two inequalities we have

$$E_p[\chi_1\hat{\Delta}\Delta] \leq \sqrt{2\ln(2)(p - p^2)(g(p) - (g(p))^2 + T_1)tI_{p,S^t}} \tag{4.22}$$

If $A_2$ holds we will have $\hat{\Delta} > 0$ and $\Delta < \frac{T_2\hat{\Delta}}{T_1}$. Combining this with (4.15) we get

$$E_p[\chi_2\hat{\Delta}\Delta] \leq \frac{T_2}{T_1}E_p[\chi_2\hat{\Delta}^2] \leq \frac{T_2}{T_1}E_p[\hat{\Delta}^2] \leq \frac{\ln(2)T_2I_{p,S}}{2T_1} \tag{4.23}$$

Finally in the case $A_3$ holds we use the trivial bound $\hat{\Delta}\Delta < t$. Assume $T_2 \leq 2(p - p^2)t$, and remember that $E_p[\Delta] = 0$. We use Chernoff inequality to bound the probability of $\Delta > T_2$ and conclude that

$$E_p[\chi_3\hat{\Delta}\Delta] \leq tE[\chi_3] \leq 2te^{-\frac{T_2^2}{4(p-p^2)t}} \tag{4.24}$$

assuming $T_2 \leq 2(p - p^2)t$.

Substituting (4.22), (4.23), and (4.24) in (4.20) we obtain

$$g'(p) \leq \sqrt{\frac{2\ln(2)(g(p) - (g(p))^2 + T_1)tI_{p,S^t}}{p - p^2}} + \frac{2T_2I_{p,S^t}}{T_1(p - p^2)} + \frac{2t}{p - p^2}e^{-\frac{T_2^2}{4(p-p^2)t}} \tag{4.25}$$

given that $T_1 > 0$ and $0 < T_2 \leq 2p(1-p)t^2$. Now to make things simpler we work with the limit of function $g$ and provide an asymptotic bound. Let $\alpha = \liminf_{t\to\infty} tI_{p,S^t}$ and let $t_i$ be a sequence of positive integers such that when $i$ tends to infinity $t_i \to \infty$ and $tI_{p,S^t} \to \alpha$. In distribution $\mathcal{B}_{p,S^{t_i}}$ let $g_i(p) = E[f]$. In other words $g_i$ is the function $g$ studied above for $t = t_i$.

Inequality (4.19) implies that functions $g_i^{'}$, the derivatives of $g_i$, obey a *uniform bound*. This means there is a constant $c(p)$ which depends only $p$ and not on $i$ or $t_i$ such that for all $i$, $g_i(p) < c(p)$. Given this, we can employ Arzela-Ascoli theorem of functional analysis and conclude that there exists a subsequence of functions $g_i$ that *uniformly converges* to a function $g_\infty$. Actually we do not need all the power of this theorem, and only a weaker assertion that a subsequence of functions $g_i$ *pointwise converge* to $g_\infty$ is enough. This means that there exists a sequence $\ell_i$ such that for each $p$, $g_{\ell_i}(p) \to g_\infty(p)$ when $i$ tends to infinity.

This function $g_\infty$ will be continuous and also we have $g_\infty(0) = 0$ and $g_\infty(1) = 1$. Unfortunately $g_\infty$ may not be differentiable so instead of working with its derivative we work with the upper derivative $g_\infty^+(p) = \limsup_{h\to 0} \frac{g_\infty(p+h)-g_\infty(p)}{h}$. This upper derivative exists due to continuity of $g_\infty$. We want to rewrite (4.25) for $g_\infty$, but first we want to choose $T_1$ and $T_2$ in a way that simplifies the right hand. For different $g_i$ we may need to define $T_1 = T_{1,i}$ and $T_2 = T_{2,i}$ separately. For our purpose it's enough to make sure the following conditions are respected. When $i$ tends to infinity,

$$\frac{T_{2,i}}{\sqrt{t_i \ln t_i}} \to \infty, \ \frac{tT_{1,i}}{T_{2,i}} \to \infty \text{ and } T_{1,i} \to 0.$$

Then we will have

$$g_\infty^+(p) \leq \sqrt{\frac{2\ln(2)\alpha(g_\infty(p) - (g_\infty(p))^2)tI_{p,S^t}}{p-p^2}} \tag{4.26}$$

Let $G(p) = \arccos(1 - 2g_\infty(p))$ for $p \in [0,1]$. We have $G(0) = 0$ and $G(1) = \pi$ and $G$ is continuous at 0,1. Also $g_\infty(p) = (1-\cos(G(p)))/2$ and $\sin(G(p)) = 2\sqrt{((g_\infty(p)-(g_\infty(p))^2))}$. Therefore

$$g_\infty^+(p) = \sin(G(p))G^{'+}(p)/2 = \sqrt{(g_\infty(p) - (g_\infty(p))^2)}G^{'+}(p),$$

where $G^{'+}(p)$ denotes the upper derivative of $G(p)$. Using (4.26) we conclude

$$G^{'+}(p) \leq 2\ln(2)\alpha(p-p^2)^{-1/2}$$

for $0 < p < 1$. Then we have

$$\pi = G(1) - G(0) \leq \int_0^1 2\ln(2)\alpha(p - p^2)^{-1/2}dp = 2\ln(2)\alpha\pi.$$

So we must have $\alpha \leq \frac{1}{2\ln 2}$.■

## 4.6 Support of $D^t$

In order to have an explicit construction of the optimal fingerprinting code described, an explicit construction of the distributor's optimal distribution, $D^t$, is required[3]. Unfortunately we don't know much about this distribution. We have only been able to prove some bounds on the size of the support of $D^t$. The theorem below states this result.

**Theorem 16** *Let $K_t$ denote the size of the support of $D^t$, the optimal bias distribution. Then*

$$\sqrt{\frac{t}{2\ln 2\log t}} \leq K_t \leq \frac{5}{2}t - 1$$

Before proving the theorem formally, we present the idea of the proof informally. The lower bound is proved based on the following idea. Assume that support of $D^t$ is too small. Then there is an interval $A$ of size at least $\epsilon$, for a properly chosen $\epsilon$, such that $A \subset [0, 1]$ and $A$ excludes all points in the support of $D^t$. Then we can construct a pirate strategy $S$, such that for any $p \notin A$, $I_{p,S} < V_t$. Given the definition of $V_t$, this is a contradiction.

As for the upper bound we consider an equation whose roots include all points in the support of $D^t$, and possibly some other points. This equation has log factors, so we go through some algebraic manipulation to transform it into an equation involving polynomials. Then the degree of this polynomial along with the number of potentially lost roots during algebraic manipulations gives us an upper bound on the size of the support of $D^t$.

**Proof:** For the lower bound let $\epsilon = \sqrt{\frac{\ln 2\log t}{t}}$ and let $\alpha \in [0, 1]$ be arbitrary. For $x \in \{0, 1\}^t$ let $S(x) = 0$ if $\frac{|x|}{t} < \alpha$ and $S(x) = 1$ if $\frac{|x|}{t} \geq \alpha$. As this channel $S$ is deterministic we have $H(f|x) = 0$ in the distribution $B_{p,S}$ for any $p$. In the same distribution we have

$$I_{p,S} = I(f; x) = H(f) - H(f|x) = H(f) = h(E[f]). \tag{4.27}$$

[3]As we mentioned before, other bias distributions may be used and the construction still works, but the rate achieved, will not be necessarily optimal.

For $p \leq \alpha - \epsilon$ we have

$$E[f] = Pr[|x| \geq \alpha t] \leq e^{-2\epsilon^2 t} = 1/t^2$$

by the Chernoff bound. Using $h(u) \leq u \log(e/u)$ we obtain

$$I_{p,S} = h(E[f]) \leq h(1/t^2) \leq \frac{\log(et^2)}{t^2} < V_t,$$

where the last inequality comes from Theorem 15.

We can prove $I_{p,S} < V_t$ similarly for $p \geq \alpha + \epsilon$. But by the definition of $V_t$ we have $E_{p \in D^t}[I_{p,S}] \geq V_t$, so there should be at least one $p \in (\alpha - \epsilon, \alpha + \epsilon)$ that belongs to the support of $D^t$. As $\alpha$ was arbitrary, this implies the claimed lower bound on the size of the support of $D^t$.

For the upper bound note that each point $p$ in the support of $D^t$ is a root of the equation $(I_{p,S^t} = V_t)$. Here equation 4.27 shows $I_{p,S^t} = h(M(p)) - K(p)$ with the binary entropy function $h$ and some degree $t$ polynomials $M$ and $P$. Thus, we have to bound the number of roots of the function

$$h(M(p)) - K(p) - V. \tag{4.28}$$

Now the first derivative of (4.28) is

$$M'(p)(\log M(p) - \log(1 - M(p))) - K'(p). \tag{4.29}$$

It is not hard to see that $M'(p)$ is not identically zero. Divide by $M'(p)$ to get

$$\log M(p) - \log(1 - M(p)) - \frac{K'(p)}{M'(p)} \tag{4.30}$$

and compute the derivative again:

$$\frac{M'(p)}{\ln 2 M(p)(1 - M(p))} - \frac{K''(p)M'(p) - M''(p)K'(p)}{M'(p)^2} \tag{4.31}$$

Multiplying by the denominators we obtain a polynomial of degree at most $4t - 3$. One can see that this is not identically zero (or it is even simpler to see that if it were zero, then (4.28) would be a polynomial of the form $cM + d$ and thus it would have at most $t$ roots as needed). Consequently (4.31) has at most $4t - 3$ roots. Given that $M'(p)$ is of degree less than $t$, we can partition $(0, 1)$ to at most $t$ intervals such that (4.30) is differentiable on each interval. On each of these intervals (4.30) has at most one root more than (4.31).

So (4.30) has at most $5t-3$ roots. These are all roots of (4.29) as well. Any extra root of (4.29) would be a common root of $M'(p)$ and $K'(p)$ and so a root of the polynomial that we get at the last step. So (4.29) has also at most $5t-3$ roots.

Each root of (4.28) is also a maximum and consequently a root of its derivative (4.29) too. In addition to these, (4.29) must have at least one root between each two roots of (4.28). As (4.29) has at most $5t-3$ roots (4.28) has at most $\frac{5}{2}t-1$ roots as claimed. ■

The theorem above was published in [3]. After this paper, Huang and Moulin [27] based on an earlier result by Moulin [34] constructed a fingerprinting code similar to ours. Using Theorem 10 it was shown in [27] that the upper bound on $K_t$ is at most $K_t \leq \lfloor t/2 \rfloor + 1$. This is based on Lemma 13 that implies strategy of second player in fingerprinting game can be confined to a subset of $\Re^{\lfloor t/2 \rfloor}$. Then Theorem 10 implies $K_t \leq \lfloor t/2 \rfloor + 1$.

# Chapter 5

# Quick Accusations

Our construction in the previous chapter achieves highest known rate for fingerprinting, but the accusation algorithm of our code is slow. In this chapter we address this problem.

In Section 5.1 we modify the accusation algorithm of the previous chapter. The new algorithm has a different analysis, specially when proving no innocent user is accused. In Subsection 5.1.1 we use the new accusation algorithm and its analysis to construct a family of fingerprinting codes that are parameterized by a tradeoff parameter $k \in \{1, 2, \cdots, t\}$. The codes with small $k$, have a faster accusation algorithm, but lower rate while the codes with large $k$ have slower accusation algorithm and higher rate.

In Section 5.2 we look at two approaches for improving the result of the Section 5.1.1. While we show the first method does not work for large $t$, it remains open to study the second approach.

In Section 5.3 we show that for two pirates we do not need to sacrifice rate to speed up the accusation algorithm. We construct an accusation algorithm which runs in $\tilde{O}(N)$ and achieves the same rate as our construction in the previous chapter.

In Section 5.4 we introduce a new model for fingerprinting called *weak fingerprinting* and a weak fingerprinting code. The reason for introducing the model is discussed there.

Not surprisingly, in this chapter we will use the notation of the last chapter. Among other things we will repeatedly work with distributions $B_{D,S}$ and $B_u$ that we defined in the previous chapter.

## 5.1 Modifying the accusation algorithm

The high rate fingerprinting scheme that we constructed achieves the highest known fingerprinting rate, but the running time is growing exponentially with $t$. A simple observation allows us to construct a family of fingerprinting schemes parameterized with an integer $k \in \{1, 2, \cdots, t\}$ that controls a tradeoff between the rate and efficiency of the accusation algorithm. This parameter is chosen by the distributor when constructing the code. Smaller values of $k$ result in a code with a lower rate and faster accusation algorithm.

The codes that are achieved at the two extremes of our tradeoff are not new. For $k = t$ we achieve the same code that was constructed in the last chapter. Independent of us a similar code with the same rate has been constructed in [27] building on the results of [34]. Interestingly, our construction here is different from both previous constructions. These three different constructions, achieve the same rate and have similar code generation algorithms but accusation algorithm and its analysis is different. When proving that almost surely some pirate will be accused, our analysis in this chapter is almost the same as the one in the previous chapter. When proving no innocent user will be accused these three constructions are different. The code on the other end of extreme ($k = 1$) is also constructed in [26] building on the results of [34]. Our algorithm and analysis in this chapter is different from that of [26].

For any $k$ the running time of the accusation algorithm is $\tilde{O}(N^k)$. For $k = 1$ we achieve linear running time which matches the running time of [44]. While the rate of our code with $k = 1$ will be smaller than the rate of our previous code, it is better than the rate of [44] and subsequent improvements that we are aware of. We will discuss this issue soon.

In the rest of this section we introduce a new accusation algorithm for the high rate code of the last chapter. At the end we make an observation that results in the tradeoff. The result on tradeoff will be discussed in section 5.1.1.

We start by stating a few definitions and then the new accusation algorithm. Then we will prove the correctness of the new accusation algorithm.

In the last chapter we defined the notion of *perceived strategy* for a group of users which is the conditional type of the forged copy given user codewords. In this chapter we use a very similar concept. For a tuple $u$ of users, let $S_u$ denote the conditional type of a forged copy given codewords of users in $u$. This is almost identical to the notion of perceived strategy, except that the former is used for a tuple of users $u$ when we suspect that $u$ includes all

pirates and no innocent users. $S_u$ is a more general concept and we use it for any tuple $u$ of users.

Also in this chapter we use two distributions $B_u$ and $B_{p,S}$ in this generalized setting. If for $u \subset w$, $|u| < |w| = t$ then $B_u$ is the marginal distribution of $B_w$ with users in $w - u$ removed.

Our counterpart of $B_{p,S}$ in this chapter is a little bit more complicated. If tuple $u$ is of size $t$ then $B_{p,S_u} = B_{p,S}$ where $S$ is the perceived strategy of $u$. Also for any $v \subset u$, $B_{p,S_v}$ is the marginal distribution of $B_{p,S_u}$ with users in $v - u$ removed. Now for $v$, $|v| < t$ where the corresponding superset of size $t$ is unknown, we can still define $B_{p,S_v}$ similar to $B_{p,S}$ except that now we have a tuple of size smaller than $t$, and instead of a perceived strategy we work with $S_v$ which is a generalization of the notion of perceived strategy[1]. For a given tuple $u$ of size $t$ and $v \subset u$, two definitions of $B_{p,S_v}$, presented in this paragraph, are equivalent as the reader can easily verify. Similar to the last chapter we can extend this definition and define $B_{D,S_v}$ where $D$ is a bias distribution.

The following definition is inspired by the concept of **Maximum Penalized Mutual Information (MPMI) decoder** of [34].

**Definition:** In a fingerprinting code with rate $R$ that has a bias based code generation let $\textbf{Penalty}(u) = I(x_u; f|p) - |u|R$. Tuple $u$ is called **high penalty**, if for any $w \subset u$, $\textbf{Penalty}(u) \geq \textbf{Penalty}(w)$. The mutual informations implicit in this definitions are interpreted in $B_{p,S_u}$.

Now we are ready to state and prove the correctness of our new accusation algorithm.

**Accusation Algorithm:** Given a forged copy $f$, for a $j$-tuple $u$ of users ($j \leq t$) do the following. If variation distance of $B_u$ and $B_{D,S_u}$ is larger than $\delta$, then accuse no one in $u$. (Some members of $u$ may be accused later in the course of the algorithm). Otherwise if $u$ is high penalty and has positive penalty accuse all members of $u$. Repeat this algorithm for all $j$-tuples of users ($j \leq t$).

Now we need to define some notation. We define two conditional distributions $T$ and $T_{u,w}$. More specifically, let $u$ be a $j$-tuple of users. Let $w$ be a tuple of users as well. Now let $T_{u,w}$ denotes the conditional type[2] of $u$ given the forged copy $f$, codewords of users in $w$ and bias $p$. Also let $T$ be a conditional distribution (i.e. a matrix) defined as follows. Let rows of $T$ be indexed by $2^j$ different values that a $j$-tuple $u$ of users may observe in a given

[1]The notion of perceived strategy was defined for a set of $t$ users.

[2]Remember that any type is a distribution as well.

column of code-matrix. Also let columns of $T$ be indexed by $p \in [0,1]$ where $p$ belongs to the support of bias distribution. Now $T$ is a conditional distribution that given the bias in a column, gives the probability that a $j$-tuple of users $u$, in the second phase of bias based code generation, receive a given sequence of $j$ bits in that column. The reader may notice that $T$ depends on size of $u$, but our notation does not show this. Tuple $u$ will be clear from the context so we drop it in our notation.

Shortly we will need to compare $T$ and $T_{u,w}$ for given tuples $u$ and $w$. For this comparison to make sense, we would like to *pretend* that $T$ depends on $f$ and $w$ as well. For this purpose, each column of $T$ will be replicated $2^{|w|+1}$ times and each copy will be indexed by one binary sequence of length $|w|+1$. Each such value represents the value of $(x_w, f)$ in a column of codematrix. From now on, whenever we talk about $T$, we mean $T$ with replicated columns. Again our notation does not show the dependency of $T$ on size of $w$ as $w$ is clear from the context.

To make this definition clear we explain it in more detail for special case of $j = 1$ and single bias $p$ and $|w| = 0$. In this case each of $T$ and $T_{u,w}$ is a two by two matrix, in which rows and columns are labeled by members of $\{0,1\}$. Then column $i$ of $T_{u,w}$ is the type of user $u$ given $f = i$. For $T$ things are easier. Conditional on the bias being $p$, all columns of $T$ are equal and

$$T_{1i} = p, T_{0i} = 1 - p.$$

In the last chapter, when proving the correctness of our accusation algorithm, we were working with two different interpretations of mutual information, depending on the underlying distribution. In this section our analysis is different and we use Theorem 6. So instead of mutual information we work with relative entropy. As we will see below, this relative entropy is equal to a variation of mutual information that is different from both interpretations used in the last chapter, but close to them. The crucial point that allows us to use this variation is that for a tuple $u$ of pirates, if $B_u$ and $B_{D,S_u}$ have variation distance zero, then all three versions of mutual information have the same value.

For tuples $u, w$ of users (potentially including pirates) we are interested in the value of $D(T_{u,w}||T|f, x_w, p)$. To calculate this value we look at the bias vector, codewords of users in $u$ and $w$ and the forged copy. Then the corresponding conditional relative entropy is calculated as defined in equation (2.3). Here $(f, x_w, p)$ is a random variable distributed according to $B_w$

For now we work with a single bias $p$. Generalizing our observation below to an arbitrary

bias vector is straightforward. Also in our calculations below we assume $u$ is of length one. Calculations for arbitrary size $u$ is similar. Let $q_1 = p$ and $q_0 = 1 - p$. Then

$$\begin{aligned} D(T_{u,w}||T|f, x_w) &= \sum_{(i,j)\in\{0,1\}^{|w|+2}} Pr[(f, x_w) = j]Pr[x_u = i|(f, x_w) = j] \log \frac{Pr[x_u = i|(f, x_w) = j]}{q_i} \\ &= \sum_{j\in\{0,1\}^{|w|+1}} -Pr[(f, x_w) = j]H(x_u|(f, x_w) = j) \\ &\quad + \sum_{(i,j)\in\{0,1\}^{|w|+2}} Pr[(f, x_w) = j]Pr[x_u = i|(f, x_w) = j] \log \frac{1}{q_i} \\ &= - H(x_u|f, x_w) + Pr[x_u = 1] \log \frac{1}{p} + Pr[x_u = 0] \log \frac{1}{1-p} \end{aligned} \tag{5.1}$$

Let's denote the value of 5.1 by $\hat{I}(x_u; x_w|p)$. The quantity $-H(x_u|f, x_w)$ above is interpreted in $B_{u\cup w}$ as it is clear from the context. The next term is a bit strange though. In $Pr[x_u = 1] \log \frac{1}{p} + Pr[x_u = 0] \log \frac{1}{1-p}$ both probabilities are calculated in $B_u$, but $p$ is the probability of $x_u = 1$ in $B_{D,S_u}$. If $B_u$, and $B_{D,S_u}$ have variation distance zero, then value of 5.1 is equal to $I(x_u; x_w|p)$ interpreted in either $B_u$ or $B_{D,S_u}$, but when the variation distance is not zero $\hat{I}$ might be different from both interpretations of $I$. It is important to observe that $\hat{I}$ is a continuous function of each of its parameters as well.

Now we are ready to prove the correctness of our modified accusation algorithm. Without loss of generality in the proofs we concentrate on the bias distribution $D^t$ and prove that any rate arbitrarily close to $R_t$ is achievable.

**Lemma 17** *Let a group of pirates of size subexponential in $n$, perform a pirate strategy and produce a forged copy $f$. Let $u$ be a tuple of users accused by the modified accusation algorithm given $f$ as input. The probability that $u$ includes any innocent user is exponentially small.*

**Proof:** Similar to Lemma 12, the mutual informations in the accusation algorithm are done in $B_{D,S_u}$, but we need to prove that our claim holds for $\hat{I}$. If the variation distance of $B_u$ and $B_{D,S_u}$ is small then we have

$$\hat{I}(x_u; f|p) \geq I(x_u; f|p) - \delta_1. \tag{5.2}$$

Now let's assume $u$ can be partitioned to $u_1$ and $u_2$ such that

- Members of $u_1$ are pirates and members of $u_2$ are innocent.

- It is possible that $u_1 = \emptyset$, but $u_2$ is not empty.

First we show that the probability of the following event

$$I(x_{u_2}; f, x_{u_1}|p) \geq |u_2|R_t \tag{5.3}$$

is exponentially small. Here we assume without loss of generality that bias distribution is $D^t$ which was introduced in the last chapter.

Putting Inequalities 5.3 and 5.2 together we conclude $\hat{I}(x_{u_2}; f, x_{u_1}|p) \geq |u_2|R_t - \delta_1$ This means $D(T_{u_2,u_1}||T|f, u_1, p) \geq |u_2|R_t - \delta_1$. Theorem 6 implies that this happens with probability no more than $2^{-n(|u_2|(R_t-\delta_1))}$. There are $2^{n|u_2|R}$ choices of $u_2$ and a subexponential number of choices for $u_1$ (if any). Setting $R \leq R_t - \delta_1$ and taking a union bound is enough to prove our claim.■

**Remark:** The reason that we need to keep working with distribution $B_{D,S}$ and its variations is that we want to build our result on top of the fingerprinting game. Otherwise we can make the above proof simpler by working in $B_u$.

**Lemma 18** *Let a group of at most t pirates perform a pirate strategy and produce a forged copy f. Let z be a tuple of some (but not necessarily all) pirates. Assume z includes no innocent user. Then with high probability the distributions $B_z$ and $B_{D,S_z}$ have variation distance less than $\delta$.*

**Proof:** In Lemma 11 we proved that for the set of all pirates this claim holds i.e. two distributions $B_u$ and $B_{p,S}$ have variation distance less than $\delta$, where $u$ is the set of all pirates. Now $B_z$ and $B_{p,S_z}$ are marginal distributions of $B_u$ and $B_{p,S}$, so their variation distance cannot be larger than $\delta$.■

**Theorem 19** *Let a group of at most t pirates perform a pirate strategy and produce a forged copy f. With high probability the output of the modified accusation algorithm is a non-empty set of pirates that includes no innocent user.*

**Proof:** Lemma 18 shows that any set of pirates will pass the first step of the accusation algorithm. The existence of a saddle point for fingerprinting game implies that the set of all pirates has high penalty. Any subset of this set which has the highest penalty is clearly a high penalty set. Lemma 17 implies that it is unlikey that an innocent user is accused.■

So far in this section we have only modified the accusation algorithm of the last chapter. The new accusation algorithm is simpler. In addition it has an advantage when we want to check whether a given set of size less than $t$ of users includes any pirates. Let $k$ denote the size of the given set of users. Using the algorithm of the previous chapter we needed time $\tilde{O}(N^t)$ if the given set includes any pirate or not. Using the new algorithm this can be done in time $\tilde{O}(N^k)$. This observation will be used in the next section to construct a family of codes that demonstrates a tradeoff between rate and efficiency of accusation algorithm.

### 5.1.1 The tradeoff

Now to construct our family of fingerprinting schemes that achieves the complexity-rate tradeoff we only need to modify the fingerprinting game of section 4.2.

Let $k \in \{1, 2, \cdots, t\}$ be a given integer. Paula chooses a *symmetric* channel $S$ and David chooses a bias $p \in [0, 1]$. After both players revealed their strategies, Paula pays $I(x_z; f)$ to David where $|z| = k$. The mutual information is evaluated in $B_{p,S}$. Let $w = \{1, 2, \cdots, t\}$ then $x_w$ is a tuple of $t$ bits each of them generated independently with expectation $p$. Then $z$ is an arbitrary subset of $w$ of size $k$. Given $x_w$, $f$ is generated using channel $S$. It is important that Paula is confined to symmetric channels, otherwise the formulation of the game would have been more complicated. Using Lemma 13 we will see that we do not lose generality by imposing this restriction. As in Section 4.2 we argue that Paula is always better off using a pure strategy than a mixed one. David's optimal strategy is mixed. This is a distribution over $[0, 1]$ and Lemma 13 implies that it is symmetric with respect to $\frac{1}{2}$.

**Lemma 20** *We have*

$$\max_D \min_S E_{p \in D}[I(x_z; f)] = \min_S \max_p I(x_z; f),$$

*where $p \in [0, 1]$, $S$ is an eligible symmetric channel, $D$ is a distribution over $[0, 1]$, $z$ is a subset of $\{1, 2, ..., t\}$ and $|z| = k$.*

Let $V_t^{(k)}$ denote the value of this game. In this game we know that there exists a $k$-tuple of pirates such that $I(x_z; f|b) \geq V_t^{(k)}$. For symmetric channels this is a consequence of Lemma 20. For non-symmetric channels, it is a result of convexity of mutual information.

For tradeoff parameter $k$ our code generation and accusation algorithm are as follows. Let $D^{t,k}$ denote the optimal strategy for David in the above game when tradeoff parameter is $k$.

**Code Generation:** Use bias based code generation with $D^{t,k}$ as bias distribution.

**Accusation Algorithm:** Repeat the following for all $1 \leq j \leq k$ in increasing order of $j$. Given a forged copy $f$, for all $j$-tuples $u$ of users do the following. If variation distance of $B_u$ and $B_{D,S_u}$ is larger than $\delta$, then accuse no one in $u$. (Some members of $u$ may be accused later in the course of the algorithm). Otherwise if $u$ is high penalty and has positive penalty accuse all members of $u$.

**Remark.** The increasing order of $j$ is not necessary for the correctness of the algorithm. We added it to the algorithm to simplify the presentation of some of the upcoming material.

The above algorithm has a running time of $\tilde{O}(N^k)$. For tradeoff parameter $k$ we can choose the rate of the code to be $R_t^{(k)} = \frac{1}{k} V_t^{(k)}$. Then we can prove the correctness of our construction by proving analogus versions of Lemmas 18 and 17. Since the proofs are almost identical we do not repeat it again.

Intuitively it is clear that larger $k$ results in higher rate. When $k$ is larger the accusation algorithm does everything that it does for smaller $k$ plus some extra work. This means that the rate can only increase when we increase $k$. To make this intuition more precise we consider two codes $C_1$ and $C_2$ with tradeoff parameters $k = k_1$ and $k = k_2$. Assume each code is constructed using the corresponding optimal bias distribution. Then the rate of $C_1$ is greater or equal to the rate of $C_2$ if and only if $k_1 > k_2$. To prove this claim we need the following observations.

The following is easy to verify based on definition of mutual information.

$$I(x_1, x_2, \cdots, x_k; f) = I(x_1; f) + I(x_2; f|x_1) + \cdots + I(x_k; f|x_1, \cdots, x_{k-1}). \tag{5.4}$$

Also we have

$$I(x_j; f|x_1, \cdots, x_{j-1}) = H(x_j|x_1, \cdots, x_{j-1}) - H(x_j|f, x_1, \cdots, x_{j-1}) \tag{5.5}$$

Now when $j$ increases, the first term in the right hand side of expansion 5.5 remains unchanged but the second term does not increase. The observation about the second term is a result of limiting pirates to symmetric strategies and the fact that conditioning reduces entropy. Consequently we have

$$I(x_{j+1}; f|x_1, \cdots, x_j) \geq I(x_j; f|x_1, \cdots, x_{j-1})$$

and so for $k_1 > k_2$

$$\frac{1}{k_1} I(x_1, x_2, ..., x_{k_1}; f) \geq \frac{1}{k_2} I(x_1, x_2, ..., x_{k_2}; f)$$

This is because in the expansion of the left hand side all terms on the right hand side appear plus some larger terms. This is enough to prove our claim.

The rate of our code for $k = 1$ is $I(x_1; f|p)$ where $x_1$ is generated using the distributor's optimal strategy. Also $f$ is generated using pirates' optimal strategy. This rate is better than the rate of [44] and subsequent improvements that we are aware of. The code in [44] and later improvements, all have an accusation algorithm which is based on weighted Hamming distance. The crucial point here is this. Let $x_1$ and $x_2$ be two codewords and let $T_{x_i|f,p}$ denote the conditional type of $x_i$ given $f$ and the bias vector. If $T_{x_1|f,p} = T_{x_2|f,p}$ then accusation algorithm of [44] and its improvements either accuse both users 1 and 2 or none of them. The same is true for our tradeoff code with $k = 1$.

Now if in our code the pirates use optimal strategy and we generate new codewords to increase the number of codewords to more than $2^{nR_t^{(1)}}$ then using Theorem 6 we can show that with high probability there exist two users $i$ and $j$ such that $i$ is a pirate, $j$ is innocent, and $T_{x_i|f,p} = T_{x_j|f,p}$. This means that our code cannot tolerate any rate higher than $R_t^{(1)}$. The same is true for the accusation algorithm of [44] and subsequent papers. These algorithms either accuse both $i$ and $j$ or accuse none of them. Both cases are wrong, so these algorithms cannot tolerate a rate higher than $R_t^{(1)}$ as well. Consequently for any given bias distribution, the rate of the accusation algorithm of [44] and its improvements are not higher than the rate of our code for $k = 1$.

This observation is important from the practical point of view. The optimal bias distributions for our constructions in the last chapter and this one are unknown. Also, it is hard to evaluate the rate of the code for a given distribution. This observation assures us that whatever our bias distribution is, we can achieve a code that is better than best previously known codes. This justifies practical use of our constructions.

## 5.2 A direction for improvement?

Our accusation algorithm above seems to suggest a potential approach to improve our tradeoff. Our next result on two pirates is a good example of this improvement. In this discussion we consider the general case of $t$ pirates. To simplify things we focus on the special case of $k = 2$. Choosing $k = 2$ results in a quadratic accusation algorithm with the rate $R_t^{(2)}$ which is lower than the optimal rate. Now we ask this question: can we keep this rate and speed up the accusation algorithm? We look at a potential way of achieving this

goal that works perfectly well in the case of two pirates, as we will see in the next section.

Let's modify our accusation algorithm slightly. For $k = 2$ our accusation algorithm has two rounds, in the first round we look at single users and in the second round we look at pairs of users. Our slightly modified version of the algorithm will stop after the first round, if a pirate is found in the first round. We will be able to speed up this algorithm if in the second round of the algorithm we need to consider only a small subset of the set of all users. Ideally, if this smaller set is of size $O(N^{1/2})$, the overall running time will be linear.

Can we prove a user is innocent after the first round of the algorithm? The fact that we didn't find any pirate in the first round means that for all users $i$, $I(x_i; f|p) \leq R_t^{(2)}$. In this formula biases $p$ are chosen from distribution $D^{t,2}$. On the other hand we know that with high probability the algorithm will find a pair $(i, j)$ of pirates in the second round. This means $I(x_i, x_j; f|p) > 2R_t^{(2)}$. Again $p \in D^{t,2}$. Now assume we construct the tradeoff code $k = 1$ using distribution $D^{t,2}$. The rate that we achieve will be less than $R_t^{(1)}$ which is in turn smaller than $R_t^{(2)}$. Let $\hat{R}$ denote this rate. Now the construction of the tradeoff code implies that there exists a pirate $i$ such that $I(x_i; f|p) > \hat{R}$. Moreover we know that (using the convexity of mutual information) the average of $I(x_i; f|p)$ for all pirates $i$ is at least $\hat{R}$. An important point here is that the value of $I(x_i; f|p)$ does not depend on $k$. The value of $k$ only determines the number of rounds of the algorithm that should be run and the rate of the code. Now let's go back to the code with $k = 2$ and bias distribution $D^{t,2}$ and for pirates $i \in \{1, 2, \cdots, t\}$ look at the value of $I(x_i; f|p)$. All these $t$ values are smaller than $R_t^{(2)}$ and their average is larger than $\hat{R} < R_t^{(2)}$. Now we want to answer this question: what is the minimum possible value of $I(x_i; f|p)$ for pirate $i$. A little bit of thought shows that the worst case happens when for $t - 1$ pirates the value of $I(x_i; f|p)$ is very close to $R_t^{(2)}$ and there is one pirate $j$ such that $MinI = I(x_j; f|p) = \hat{R} - (t - 1)\Delta$ where $\Delta = R_t^{(2)} - \hat{R}$. Now any user $\ell$ with $I(x_\ell; f|p) < MinI$ is definitely innocent.

So, we can prove that some users are innocent and improve the running time of the algorithm in the second round. For $t = 2$ as we will see shortly this can be done perfectly well. For slightly larger values this method may be useful and more investigation is needed. Unfortunately for large $t$, this value is not helpful as one of the following two cases will happen.

1. $R_t^{(2)} - \hat{R} = o(\frac{1}{t^2})$. If this happens our discussion here is irrelevant, as in this case choosing $k = 1$ we achieve linear accusaton algorithm with effectively the same rate

as $k = 2$.

2. $R_t^{(2)} - \hat{R} = \frac{c}{t^2}$. In this case $\Delta = \frac{c'}{t^2}$. So $MinI$ will be negative very soon as we increase $t$. This implies that for $t$ larger than a threshold this method does not help us to improve the tradeoff code with $k = 2$.

A potential approach to solve this problem is to prove that the maximum value of $I(x_i, x_j; f|p)$ for pirates $i, j$ is achieved by the pirates with two highest values of $I(x_i; f|p)$. Or at least we say if we lose a certain fraction pirates when discarding innocent users still the remaining pirates include a pair $i, j$ such that $I(x_i, x_j; f|p) > 2R_t^{(2)}$. This guarantees that the algorithm will finish in the second round. It is an interesting open problem to study this approach further.

## 5.3 A fast accusation algorithm for two pirates

The main drawback of our high rate fingerprinting code is the slow accusation algorithm, running time of which grows exponentially in $t$, the maximum number of pirates. An ideal fingerprinting scheme will have an accusation algorithm which is independent of $t$ like those of [44, 12] while keeping the rate of our high rate fingerprinting (or improving it if possible, but we conjecture that it achieves the optimal rate). In this section we present an accusation algorithm for the special case of $t = 2$ with running time $\tilde{O}(N)$. We prove that using this new algorithm along with the code generation described in Section 4.2.1, we can achieve the rate $R_2 = 0.25$ while the running time of the accusation algorithm is improved from $\tilde{O}(N^2)$ to $\tilde{O}(N)$. Our algorithm here is based on the previous accusation algorithm. While previous algorithm checks all pairs of users for finding pirates, here we first restrict the number of all *suspicious pairs* to $O(N^{1/2})$ in time $\tilde{O}(N)$, then we run the previous algorithm on the short listed suspicious pairs.

The main idea of our new algorithm is to sort users in terms of their Hamming distance from the forged copy. Then if there is a user whose codeword is *too close* to the forged copy he must be a pirate. Otherwise we know that users whose codeword has a *large* Hamming distance to the forged copy are innocent. So we are left with a set of users whose codewords are *relatively close* to the forged copy. We prove the number of such users is $O(N^{1/2})$, and so we can run the previous algorithm while keeping the running time $\tilde{O}(N)$.

### 5.3.1 A useful lemma

Without loss of generality, we may assume that first codewords of the pirates are generated and pirates generate the forged copy $f$. Then the distributor generates codewords of other users. We want to upper bound the probability that at least one innocent user has relative Hamming distance smaller than $\delta < 1/2$ to $f$. Fix $\rho < \delta$. First we try to calculate the probability that one innocent user $x$ has relative Hamming distance $\rho$ from $f$. If this is the case, the bitwise xor of $x$ and $f$ denoted $x+f$ will satisfy $|x+f| = \rho n$ . Since $f$ is fixed and $x$ is chosen uniformly at random, $x+f$ is also distributed uniformly at random. All binary strings $y$ with $|y| = \rho n$, are of the same type. The size of the corresponding type class is $2^{nh(\rho)}$. So the probability of choosing $x$ such that $|x+f| = \rho n$ is $2^{-n(1-h(\rho))}$

**Lemma 21** *Let $y$ be a fixed string of length $n$. Let $x$ be string chosen uniformly at random from among all strings of length $x$. The probability that $x$ has relative Hamming distance less than $\delta < 1/2$ from $y$ is at most $2^{-n(1-h(\delta))+O(\log n)}$.*

### 5.3.2 Accusation algorithm

Now given Lemma 21 we modify the accusation algorithm of our high rate fingerprint code such that the rate remains equal to 0.25 and the running time of the accusation algorithm becomes linear.

**Algorithm:** In the first step of the new algorithm we check users with Hamming distance smaller than $0.207n$ to the forged copy. If there is more than one such user algorithm finishes by declaring *failure*. If there is exactly one such user it will be accused as pirate and the algorithm finishes. Otherwise we look at all users whose Hamming distance with the forged copy is less than $0.293n$. We will show that the number of such users is $O(N^{1/2})$. Now we run the original accusation algorithm of our high rate fingerprint code on this group of users. It is easy to see that the running time of the accusation algorithm is $\tilde{O}(N)$.

**Theorem 22** *The above accusation algorithm for 2-fingerprinting finds at least one pirate in time $\tilde{O}(N)$ whp. The probability of accusing an innocent user or returning no pirate is exponentially small.*

**Proof:** Now we prove that the algorithm works correctly with high probability. Lemma 21 implies that the the probability that an innocent user has Hamming distance smaller than

$T = 0.207n$ to the forged copy is not larger than $2^{-0.26n}$. Since there are at most $2^{0.25n}$ users in the system, with high probability no innocent user will be accused in this step. Also the probability that more than one user (innocent or pirate) are closer than $T$ to the forged copy is negligible. The only remaining case is when no user has Hamming distance smaller than $T$ to the forged copy.

The Hamming distance between two pirates is $(\frac{1}{2} + o(1))n$ with high probability. This, along with marking assumption, implies that with high probability the average Hamming distance of pirates from the forged copy is $(\frac{1}{4} + o(1))n$. This implies that the distance of one pirate to the forged copy lies in the interval $[T, (\frac{1}{4} + o(1))n]$ and the distance of the second pirate to the forged copy lies in the interval $[(\frac{1}{4} + o(1))n, 0.293n]$. So if we run the original accusation algorithm of the high rate code on users with Hamming distance $\leq 0.293n$ to the forged copy, we will find at least one pirate with high probability. Lemma 21 implies that the expected number of such users is not more than $O(2^{0.123}n) = O(N^{1/2})$ with high probability. This completes the proof.

## 5.4 Weak fingerprinting

In this section we introduce a model for fingerprinting in which the accusation algorithm gets a huge help: when searching for one of the pirates it learns from an "oracle" the identity of all the other pirates. See the formal definition below. It is not surprising that we can adapt our fingerprint codes to this setup and increase their rate slightly. This model was first introduced in [3]. An improvement of the strong converse theorem of [7] by Gabor Tardos [45] implied that our method achieves weak fingerprinting capacity. Our main reason for introducing this model was to provide an evidence for our conjecture that our construction of the last chapter achieves fingerprinting capacity.

**Definition:** A weak fingerprinting code over alphabet $\Sigma$ for users $U$ consists of a *distribution algorithm*, an *oracle* and an *accusation algorithm*. The distribution algorithm is the same as in the standard model: a randomized procedure for producing codewords $X^v \in \Sigma^n$ for users $v \in U$ (and possibly some side information). A set $T \subseteq U$ of pirates use a pirate-strategy to output a forged codeword. We allow the same type of pirate-strategies as in the standard model, namely the ones obeying the marking assumption. The oracle is a function (algorithm) that has full access to the output of the distribution algorithm and the forged codeword and also to the set $T$ of pirates. It picks a pirate $u \in T$ to chase and outputs

the remaining pirates $T \setminus \{u\}$. Finally the accusation algorithm has access to the forged codeword and the outputs of the distribution algorithm and the oracle and outputs a single accused user. The accusation algorithm errs if its output is not the pirate $u$ picked by the oracle. Note that as the oracle is part of the fingerprinting code it collaborates with the accusation algorithm: it will choose the pirate $u$ that is easiest to catch.

A weak fingerprint code is *$\epsilon$-secure against $t$ pirates* if for any set $T$ of pirates of size at most $t$ and for any pirate-strategy the probability of error is at most $\epsilon$. The *rate* of fingerprint code and *weak fingerprinting schemes* are defined as in the standard model. The *weak $t$-fingerprinting capacity* is the maximal rate achievable by a a $t$-secure weak fingerprinting scheme.

To contrast with weak fingerprinting we call the fingerprinting as defined in Section 4.2 the *standard model*. As in the standard model we can allow the accusation algorithm of the weak fingerprinting to stop without accusing any user and consider this type of failure less serious than falsely accusing an innocent user. The codes constructed in this section have the advantage that for any subexponential size set of pirates using any pirate-strategy and even for any malicious oracle outputting any subset of them the probability of falsely accusing anybody is still small.

It is clear that any standard fingerprinting code can be adapted to fit this model without changing its rate or the error probability. Simply make the accusation algorithm ignore the output of the oracle and make the oracle choose the pirate that will be accused.

In the rest of this section we adapt our fingerprint codes constructed in Section 4.2 to the weak model while increasing their rate. Both the minimax result (Lemma 23) the construction is based on and the correctness (Theorem 24) are proved in the same way as in the standard model, so we omit these proofs.

For $i \in [t]$, $p \in [0,1]$ and $S$ an eligible channel we define $I^{(i)}_{p,S}$ to be the conditional mutual information $I(x_i; f|x_{(i)})$ in the probability space $B_{p,S}$. Here $x_{(i)} := (x_1 ... x_{i-1} x_{i+1} ... x_t)$. One can think of $I^{(i)}_{p,S}$ as a measure of the information, one bit of the forged codeword gives on the identity of the $i$th pirate if the identity of all the other pirates had previously been known and the pirates use a channel based strategy with the channel $S$. Here we also assume that the code distribution is bias based and $p$ is the bias chosen for the particular digit we consider.

Now we consider the game in which Paula chooses a channel $S$ and David chooses a pair $(p, i)$ with $p \in [0,1]$ and $i \in [t]$. After their simultaneous choices Paula pays $I^{(i)}_{p,S}$ to

David. As in Section 4.2 we argue that Paula is always better off using a pure strategy than a mixed one. As for Lemma 13 we use symmetry to prove that David can choose an optimal mixed strategy (distribution over $(p, i)$) in which $p$ and $i$ are independent and $i$ is distributed uniformly over $[t]$.

**Lemma 23** *We have*

$$\max_D \min_S E_{p\in D, i\in U_t}[I^{(i)}_{p,S}] = \min_S \max_{p,i} I^{(i)}_{p,S},$$

*where $p \in [0, 1]$, $i \in [t]$, $S$ is an eligible channel, $D$ is a distribution over $[0, 1]$ and $U_t$ is the uniform distribution over $[t]$.*

We use $S^t_w$ to denote a channel $S$ minimizing the right hand side and $D^t_w$ to denote a distribution $D$ maximizing the left hand side. We let $W_t$ stand for the common value of this minimum and maximum.

Let $F_{t,R,\delta,n}$ be the following weak fingerprint code. The parameters have the same meaning as in $E_{t,R,\delta,n}$. For simplicity we assume here that the number of pirates is exactly $t$. As the accusation algorithm learns the number of pirates anyway, it is easy to remove this simplifying assumption.

**Codeword generation:** We use bias based codeword generation with the bias distribution $D^t_w$. The codelength is $n$ and we choose a set $U$ of users of cardinality $\lfloor 2^{Rn} \rfloor$.

**Oracle:** Let $u = (u_1, \ldots, u_t)$ be the $t$-tuple of pirates and let $S$ be their perceived strategy. By the definition of $W_t$ and $D_t$ there exists an index $i$ satisfying $E_{p\in D_t}[I^{(i)}_{p,S}] \geq W_t$. The oracle chooses such an index $i$ and outputs the set of all pirates except $u_i$.

**Accusation algorithm:** Let $T' = \{u_1, \ldots, u_{t-1}\}$ be the output of the oracle. For any user $u_t \in U \setminus T'$ we consider the tuple $u = (u_1, \ldots, u_{t-1}, u_t)$ and the perceived strategy $S$ of $u$. We accuse $u_t$ if $S$ is eligible, the total variation distance between the distributions $B_u$ and $B_{D_t,S}$ is at most $\delta$ and $E_{p\in D_t}[I^{(t)}_{p,S}] \geq W_t$. In case more than a single user $u_t$ satisfies these conditions the accusation algorithm fails.

**Theorem 24** *For any $t$ and $R < W_t$ there exists a positive $\delta$ such that the fingerprinting codes $E_{t,R,\delta,n}$ for $n \geq 1$ form a $t$-secure fingerprinting scheme of rate $R$.*

# Chapter 6

# Open problems

In this chapter we review three major direction of research on fingerprinting that has interesting problems to work on.

## 6.1 Capacity

In this thesis we have constructed fingerprinting codes that achieve the best known rates. We conjecture that our construction of chapter 4 achieves the fingerprinting capacity. This is one of the major open problems in fingerprinting research. Actually, this problem is resolved in [34] for some other models of fingerprinting, but the proof, as is, does not work for marking assumption. It is interesting to explore what exactly can be proven for marking assumption model, using the tools introduced in [34]. Even if the proof does not work for marking assumption, it may work for relaxations of marking assumption model in which pirates can detect up to a $\delta$ fraction of columns in which they all have the same value.

The relaxations on marking assumption are considered by many researchers to be a more natural model for fingerprinting. If pirates decide to change a fragment of the document they may be able to violate marking assumption without actually finding the column in which they all have the same bit. Our constructions in this thesis work for this relaxed version of marking assumption as well. The rate that we achieve will decrease when we increase $\delta$.

## 6.2 Better constructions

Another major open problem is to improve upon results of chapter 5 of this thesis. It is not clear if the tradeoff between efficiency and rate is a result of our method or a natural restriction in fingerprinting.

We have shown that for two pirates the tradeoff is not needed and the highest known rate (conjectured to be capacity) can be achieved along with a linear time accusation algorithm.

We also have some preliminary results on three pirates that suggest tradeoff of chapter 5 is not the best possible for three pirates. One can observe that in the case of three pirates and $k = 3$ the average Hamming distance of pirates to the forged copy is smaller than the average Hamming distance of innocent users and the forged copy. So when we look at triplets of users in the accusation algorithm for $k = 3$, we know that at least one of the pirates should be from the set of users with small Hamming distance to the forged copy. This reduces the overall running time slightly.

All in all it seems that for small $t$ our tradeoff code is not the best. For larger $t$ this question is open. As we saw before, a direct generalization of our method for two pirates will not be successful for larger $t$, but the second approach that we suggested in section 5.2 needs further investigation.

There are two other problems that we are investigating but they are not completely done, so they have not been reported in this thesis. We have shown that Hamming distance plays an important role in improving the tradeoff for two and three pirates. Also, one may be curious to study minimum Hamming distance as simplest accusation algorithm.

Here the problem is to devise a fingerprinting code with one of the following accusation algorithms.

- Accuse the user with smallest Hamming distance to a given forged copy.
- Accuse all users with Hamming distance $< Z$ to a given forged copy for a properly chosen threshold $Z$.

As we discussed in chapter 2, there is little work on this problem in the case of binary alphabet. For larger alphabets the problem has been investigated in traitor tracing research.

We have considered the first kind of accusation algorithm above and our preliminary results on this problem, suggests that we can construct a code with rate $\Theta(\frac{1}{t^2 \log^2 t})$. This is

interesting because using such a simple accusation algorithm we can achieve a rate which is close to the capacity.

Another problem is studying performance of our constructions for larger alphabets. Our preliminary inspections suggest that using larger alphabets improves the rate of the code. A precise analysis of the change in the rate is not yet done.

## 6.3 Better evaluation of rates

We have shown that the rate of our tradeoff code, drops when we decrease $k$. Despite this, the rate is stated in terms of formulas involving mutual information. Precise evaluation of these formulas is an interesting and important problem.

We know that even when $k = 1$ the rate is $\frac{c}{k^2}$. In [26] authors have shown that an argument similar to theorem 14 proves similarly that $c \geq 0.29$. The important question is to determine the actual value of $c$.

At this point we don't have any reason that the difference of rate in our tradeoff code for $k = 1$ and $k = t$ is not of $o(\frac{1}{t^2})$. Indeed it will be a very interesting result to prove that the difference is $o(\frac{1}{t^2})$. Then we no longer need to use $k > 1$. Using $k = 1$ we are close enough to capacity and we have a linear time algorithm! Even if this best case scenario is not true, it is improtant to know the difference of rate between codes corresponding to different values of $k$.

# Bibliography


[1] M. Alekhnovich, E. Ben-sasson, Linear upper bounds for random walk on small density random 3-CNF, In *Proceedings of 44th Symposium on Foundations of Computer Science* , page 352, 2003.

[2] E. Amiri, E. Skvortsov, Pushing Random Walk Beyond Golden Ratio, CSR 2007, also in Lecture Notes in Computer Science, 4649, pages 44-55, 2007.

[3] E. Amiri, G. Tardos, High rate fingerprinting and fingerprinting capacity, In *Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 336-345, 2009.

[4] R. J. Anderson, ed. Information Hiding: First International Workshop, *Lecture Notes in Computer Science,* vol. 1174, 1996.

[5] R. J. Anderson, F.A.P. Petitcolas, On the Limits of Steganography, *IEEE Journal on Selected Areas in Communications,*16 (4), 1998, Page: 474.

[6] N.P. Anthapadmanabhan, A. Barg, Random Binary Fingerprinting Codes for Arbitrarily Sized Coalitions, *Proceedings of IEEE International Symposium on Information Theory* , ISIT 2006, pages 351-355, 2006.

[7] N.P. Anthapadmanabhan, A. Barg, I. Dumer, Fingerprinting capacity under the marking assumption. Submitted to IEEE Transaction on Information Theory - Special Issue on Information-theoretic Security. Available from arXiv:cs/0612073v2. Preliminary version appeared in the Proceedings of the 2007 IEEE International Symposium on Information Theory, (ISIT 2007), 2007.

[8] A. Barg, G. R. Blakely, G. Kabatiansky, Digital fingerprintg codes: Problem statements, constructions, Identification of traitors. *IEEE transactions on information theory* vol. 49, No. 4, pages 852-865, April 2003.

[9] G. R. Blakely, G. Kabatiansky, Random coding technique for digital fingerprinting codes: fighting two pirates revisited. ISIT 2004, page 202.

[10] O. Blayer, T. Tassa, Improved versions of Tardos' fingerprinting scheme. Submitted.

[11] D. Boneh, M. Naor, Traitor Tracing with Constant Size Ciphertext, In *Proceedings of the 15th ACM Conference on Computer and Communication Securit*, pages 501-510, 2008.

[12] D. Boneh, J. Shaw, Collusion-secure fingerprinting for digital data, *IEEE Transactions of Information Theory* **44** (1988), 480–491.

[13] H.G. Burchard, D.F. Hale, Piecewise polynomial approximation on optimal meshes, *Journal of Approximation Theory* **14** (1975), 128–147.

[14] R. D. Cameron, A case study in SIMD text processing with Parallel Bit Streams - UTF-8 to UTF-16 Transcoding, In *Proceedings of the 2008 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 91-98, 2008.

[15] R. D. Cameron, E. Amiri, K. Herdy, D. Lin, T. Shermer, F. Popowich, Parallel parsing with bitstream addition: An XML case study. Technical Report No. SFU-CS-2010-11, Simon Fraser University, October 26, 2010.

[16] H. Chor, A. Fiat, M. Naor, Tracing traitors, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference*, Also in *Lecture Notes in Computer Science 839, pages 257-270*. Springer 1994.

[17] R. D. Cameron, K. Herdy, E. Amiri, Parallel Bit Stream Technology as a Foundation for XML Parsing Performance. *In Proceedings of the International Symposium on Processing XML Efficiently: Overcoming Limits on Space, Time, or Bandwidth*. Balisage Series on Markup Technologies, vol. 4 (2009). doi:10.4242/BalisageVol4.Cameron01.

[18] T. Cover, J. Thomas, Elements of information theory, Wiley-InterScience, 2006.

[19] I. Csiszár, The method of types, in *IEEE Transactions on Information Theory*, vol. 44, No. 6, 1998.

[20] I. Csiszár, J. Körner, Information Theory: Coding Theorems for Discrete Memoryless Systems, Academic Press, London, 1981.

[21] I. Dumer, Equal-Weight Fingerprinting Codes, Second Interntational workshop on codeing and cryptology, *Lecture Notes in Computer Science*, Vol. 5557, pages 43-51. 2009.

[22] U. Feige, D. Vilenchik, A local search algorithm for 3SAT, Technical Report MCS 04-07, Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2004.

[23] A. Flaxman, A Spectral Technique for Random Satisfiable 3CNF Formulas, In *Proceedings of the Thirteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 357-363, 2003.

[24] H. D. L. Hollman, J. H. van Lint, J. P. Linnartz, L. M. G. M. Tolhuizen, On Codes With Identifiable Parent Property, *Journal of Combinatorics Theory A.*, Vol. 82, 1998, pages 121-133.

[25] N. J. Hopper, Toward a Theory of Steganography, PhD Thesis, School of Computer Science, Carnegie Mellon Univeristy, 2004.

[26] Y.W. Huang, P. Moulin, Capacity Achieving Fingerprint Decoding, In *Proceedings of 2009 IEEE Workshop on Information Forensics and Security*(WIFS 2009), London, United Kingdom, December 2009, pages 51-55.

[27] Y.W. Huang, P. Moulin, Saddle Point Solution of the Fingerprinting Capacity Game Under the Marking Assumption, in 2009 IEEE International Symposium on Information Theory (ISIT 2009), Seoul, Korea, June 2009.

[28] W. Jiang, Y. Li, X. Yu, Maximum IPP Codes of Length 3, *Annals of Combinatorics*, Vol. 13, 2010, pages 491-510.

[29] N. F. Johnson, S. Katzenbeisser. A Survey of Steganographic Techniques, Chapter 3 in S. Katzenbeisser (ed.), F. A. P. Petitcolas (ed.) Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Books, 2000.

[30] G.C. Kessler, An Overview of Steganography for the Computer Forensics Examiner, Forensic Science Examinations, Volume 6, Number 3,July 2004.

[31] E. Kawaguchi, R.O. Eason, Principle and Applications of of BPCS-Steganography

[32] R. J. Leonard, From Parlor Games to Social Science: Von Neumann, Morgenstern, and the Creation of Game Theory 1928-1944, *Journal of Economic Literature*, Vol. 33, No. 2 (Jun., 1995), pp. 730-761

[33] S.C. Lin, M. Shahmohammadi, H. El Gamal, Fingerprinting with minimum distance decoding, CoRR abs/0710.2705, (2007)

[34] P. Moulin, Universal Fingerprinting: Capacity and Random-Coding Exponents. arXiv:0801.3837v2, 2008.

[35] von Neumann, J: Zur Theorie der Gesellschaftsspiele, *Math. Annalen* 100 (1928) 295-320.

[36] H. Nikaido, On von Neumann's minimax theorem, *Pacific Journal of Mathematics* **4** (1954), 65–72.

[37] F. A. P. Petitcolas, R. J. Anderson, M.G. Kuhn, Infomation Hiding - A Survey, In the *Proceedings of the IEEE*, special issue on protection of multimedia content, 87(7):1062-1078, July 1999.

[38] L.A. Petrosjan, N. A. Zenkevich, Game Theory, World Scientific Publishing, 1996.

[39] J. Ville, Sur la théorie générale des jeux ou intervient l'habilité des joueurs. Traité du calcul des probabilites et de ses applications, IV, 2 (1938), 105–113, Eds. E. Borell et al., Gauthier-Villars, Paris, 1938, Vol. IV, Fascicule 2, pp. 105–113.

[40] R. Safavi-Naeini, Y. Wang, Sequential Traitor Tracing, In *IEEE Transactions on Information Theory*, Vol. 49(5), pages 1319-1326, May 2003.

[41] H.G. Schaathun, Fighting Two Pirates,In *International Symposium on Applied Algebraic Algorithms and Error Correcting Codes*, Also in Lecture Notes in Computer Science Vol. 2643, pages 71-78. May 2003.

[42] B. Skoric, S. Katzenbeisser, M.U. Celik, Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. To appear in Designs, Codes and Cryptography. Available online at `http://members.home.nl/skoric/security/symmetricTardos.pdf`.

[43] B. Skoric, T.U. Vladimirova, M. Celik, J.C. Talstra, Tardos fingerprinting is better than we thought. Available from arXiv:cs/0607131.

[44] G. Tardos, Optimal probabilistic fingerprint codes, *Journal of the ACM*, to appear. Preliminary version appeared in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, (STOC 2003), 116–125.

[45] G. Tardos, Personal communication.

[46] V. To, R. Safavi-Naini, On the Maximal Codes of Length 3 with the 2-Identifiable Parent Property, *SIAM Journal on Discrete Mathematics*, 17(4), pages, 548-570 (2004).

[47] A. L.C. Wang, An Industrial-Strength Audio Search Algorithm. In *International Symposium on Music Information Retrieval*, ISMIR 2003.

[48] `http://www.ifp.uiuc.edu/~moulin/talks/eusipco05-slides.pdf`, Retrived on October, 23, 2009.