UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Department of Telematics Engineering
Universitat Politècnica de Catalunya

Ph.D. Dissertation

On Codes for Traceability Schemes: Constructions and Bounds

José Moreira-Sánchez

Advisor:
Prof. Marcel Fernández-Muñoz

September 2013

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de qualificació de tesi doctoral

Curs acadèmic:

Nom i cognoms

Programa de doctorat

Unitat estructural responsable del programa

Resolució del Tribunal

Reunit el Tribunal designat a l'efecte, el doctorand / la doctoranda exposa el tema de la seva tesi doctoral titulada

_____

_____.

Acabada la lectura i després de donar resposta a les qüestions formulades pels membres titulars del tribunal, aquest atorga la qualificació:

☐ NO APTE ☐ APROVAT ☐ NOTABLE ☐ EXCEL·LENT

(Nom, cognoms i signatura) President/a		(Nom, cognoms i signatura) Secretari/ària
(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal

_____, _____ d'/de _____ de _____

El resultat de l'escrutini dels vots emesos pels membres titulars del tribunal, efectuat per l'Escola de Doctorat, a instància de la Comissió de Doctorat de la UPC, atorga la MENCIÓ CUM LAUDE:

☐ SÍ ☐ NO

(Nom, cognoms i signatura) Presidenta de la Comissió de Doctorat	(Nom, cognoms i signatura) Secretària de la Comissió de Doctorat

Barcelona, _____ d'/de _____ de _____

Abstract

A traceability or fingerprinting scheme is a cryptographic scheme that facilitates the identification of the source of leaked information. In a fingerprinting setting, a distributor delivers copies of a given content to a set of authorized users. If there are dishonest members (traitors) among them, the distributor can deter plain redistribution of the content by delivering a personalized, i.e., marked, copy to each user. The set of all user marks is known as a fingerprinting code. There is, however, another threat. If several traitors collude to create a copy that is a combination of theirs, then the pirated copy generated will contain a corrupted mark, which may obstruct the identification of traitors.

This dissertation is about the study and analysis of codes for their use in traceability and fingerprinting schemes, under the presence of collusion attacks. Moreover, another of the main concerns in the present work will be the design of identification algorithms that run efficiently, i.e., in polynomial time in the code length.

In Chapters 1 and 2, we introduce the topic and the notation used. We have also discussed some properties that characterize fingerprinting codes known under the names of separating, traceability (TA), and identifiable parent property (IPP), which will be subject of research in this dissertation.

Chapter 3 is devoted to the study of the Kötter-Vardy soft-decision decoding algorithm to solve a variety of problems that appear in fingerprinting schemes. The concern of the chapter is restricted to schemes based on Reed-Solomon codes. By using the Kötter-Vardy algorithm as the core part of the identification processes, three different settings are approached: identification in TA codes, identification in IPP codes and identification in binary concatenated fingerprinting codes. It is also

discussed how by a careful setting of a reliability matrix, i.e., the channel information, all possibly identifiable traitors can be found.

In Chapter 4, we introduce a relaxed version of separating codes. Relaxing the separating property lead us to two different notions, namely, almost separating and almost secure frameproof codes. From one of the main results it is seen that the lower bounds on the asymptotical rate for almost separating and almost secure frameproof codes are greater than the currently known lower bounds for ordinary separating codes. Moreover, we also discuss how these new relaxed versions of separating codes can be used to show the existence of families of fingerprinting codes with small error, equipped with polynomial-time identification algorithms.

In Chapter 5, we present explicit constructions of almost secure frameproof codes based on weakly biased arrays. We show how such arrays provide us with a natural framework to construct these codes. Putting the results obtained in this chapter together with the results from Chapter 4, shows that there exist explicit constructions of fingerprinting codes based on almost secure frameproof codes with positive rate, small error and polynomial-time identification complexity. We remark that showing the existence of such explicit constructions was one of the main objectives of the present work.

Finally, in Chapter 6, we study the relationship between the separating and traceability properties of Reed-Solomon codes. It is a well-known result that a TA code is an IPP code, and that an IPP code is a separating code. The converse of these implications is in general false. However, it has been conjectured for some time that for Reed-Solomon codes all three properties are equivalent. Giving an answer to this conjecture has importance in the field of fingerprinting, because a proper characterization of these properties is directly related to an upper bound on the code rate, i.e., the maximum users that a fingerprinting scheme can allocate. In this chapter we investigate the equivalence between these properties, and provide a positive answer for a large number of families of Reed-Solomon codes. The results obtained seem to suggest that the conjecture is true.

Resumen

Un sistema de trazabilidad o de fingerprinting es un mecanismo criptográfico que permite identificar el origen de información que ha sido filtrada. En el modelo de aplicación de estos sistemas, un distribuidor entrega copias de un determinado contenido a un conjunto de usuarios autorizados. Si existen miembros deshonestos (traidores) entre ellos, el distribuidor puede disuadir que realicen una redistribución ingenua del contenido entregando copias personalizadas, es decir, marcadas, a cada uno de los usuarios. El conjunto de todas las marcas de usuario se conoce como código de fingerprinting. No obstante, existe otra amenaza más grave. Si diversos traidores confabulan para crear una copia que es una combinación de sus copias del contenido, entonces la copia pirata generada contendrá una marca corrompida que dificultará el proceso de identificación de traidores.

Esta tesis versa sobre el estudio y análisis de códigos para su uso en sistemas de trazabilidad o de fingerprinting bajo la presencia de ataques de confabulación. Otra de las cuestiones importantes que se tratan es el diseño de algoritmos de identificación eficientes, es decir, algoritmos que se ejecuten en tiempo polinómico en la longitud del código.

En los Capítulos 1 y 2 presentamos el tema e introducimos la notación que utilizaremos. También presentaremos algunas propiedades que caracterizan los códigos de fingerprinting, conocidas bajo los nombres de propiedad de separación, propiedad identificadora de padres (IPP) y propiedad de trazabilidad (TA), que están sujetas a estudio en este trabajo.

El Capítulo 3 está dedicado al estudio del algoritmo de decodificación de lista con información de canal de Kötter-Vardy en la resolución de determinados problemas que

aparecen en sistemas de fingerprinting. El ámbito de estudio del capítulo son sistemas basados en códigos de Reed-Solomon. Empleando el algoritmo de Kötter-Vardy como parte central de los algoritmos de identificación, se analizan tres propuestas en el capítulo: identificación en códigos TA, identificación en códigos IPP e identificación en códigos de fingerprinting binarios concatenados. También se analiza cómo mediante un cuidadoso ajuste de una matriz de fiabilidad, es decir, de la información del canal, se pueden encontrar a todos los traidores que es posible identificar eficientemente.

En el Capítulo 4 presentamos una versión relajada de los códigos separables. Relajando la propiedad de separación nos llevará a obtener dos nociones diferentes: códigos cuasi separables y códigos cuasi seguros contra incriminaciones. De los resultados principales se puede observar que las cotas inferiores de las tasas asintóticas para códigos cuasi separables y cuasi seguros contra incriminaciones son mayores que las cotas inferiores actualmente conocidas para códigos separables ordinarios. Además, también estudiamos como estas nuevas familias de códigos pueden utilizarse para demostrar la existencia de familias de códigos de fingerprinting de baja probabilidad de error y dotados de un algoritmo de identificación en tiempo polinómico.

En el Capítulo 5 presentamos construcciones explícitas de códigos cuasi seguros contra incriminaciones, basadas en matrices de bajo sesgo. Mostramos como tales matrices nos proporcionan una herramienta para construir dichos códigos. Poniendo en común los resultados de este capítulo con los del Capítulo 4, podemos ver que, basándonos en códigos cuasi seguros contra incriminaciones, existen construcciones explícitas de códigos de fingerprinting de tasa positiva, baja probabilidad de error y con un proceso de identificación en tiempo polinómico. Demostrar que existen dichas construcciones explícitas era uno de los principales objetivos de este trabajo.

Finalmente, en el Capítulo 6, estudiamos la relación existente entre las propiedades de separación y trazabilidad de los códigos de Reed-Solomon. Es un resultado bien conocido el hecho que un código TA es un código IPP, y que un código IPP es un código separable. Las implicaciones en el sentido opuesto son falsas en general. No obstante, existe una conjetura acerca de la equivalencia de estas tres propiedades en el caso de códigos de Reed-Solomon. Obtener una respuesta a esta conjetura es de una importancia relevante en el campo del fingerprinting, puesto que la caracterización

de estas propiedades está directamente relacionada con una cota superior en la tasa del código, es decir, con el número de usuarios que puede gestionar un sistema de fingerprinting. En este capítulo investigamos esta equivalencia y proporcionamos una respuesta afirmativa para un gran número de familias de códigos de Reed-Solomon. Los resultados obtenidos parecen sugerir que la conjetura es cierta.

Acknowledgments

My first and most earnest acknowledgment goes to my advisor, Prof. Marcel Fernández, for his guidance, unwavering support and encouragement over the years that I spent in my dissertation. This work would not have been possible without his great expertise and involvement. Also, I would like to thank Prof. Miguel Soriano, for his support over these years, and for giving me the opportunity to join the Information Security Group, within the Department of Telematics Engineering.

I am very thankful to the dissertation committee members, Prof. Maria Bras-Amorós, Prof. Josep Cotrina-Navau and Prof. Josep Domingo-Ferrer, for their effort in reading and evaluating this dissertation.

It has been an honor and a privilege to have worked with a leading expert in the field of coding theory, Prof. Grigory Kabatiansky, whose contributions are always a source of inspiration. Also, I wish to thank our collaborators from the Department of Communications, at the Universitat Politècnica de València: Maria de los Ángeles Simarro-Haro, Prof. Francisco José Martínez-Zaldívar and Prof. Alberto González.

My gratitude is also addressed to all the members of the Information Security Group, and to my colleagues within the Department. I wish to extend a special note of thank to Dr. David Rebollo-Monedero, who has been an inexhaustible source of knowledge, with whom I have spent many lunches and enriching nonstandard conversations, and who has the ability to transform the answers to my questions into invaluable master classes. Thanks also, for their administrative and technical support, to all the staff within the Department.

Last, but certainly not least, my most special thanks goes to my parents, Isabel and Ángel, to my family, and to my closest friends.

Financial Support

This work has been financially supported, in part, by the Spanish Government through projects TEC2008-06663-C03-01 (P2PSec), Consolider Ingenio 2010 CSD2007-00004 (ARES) and TEC2011-26491 (COPPI), and by the Catalan Government under Grant 2009 SGR-1362. The author has also been awarded an FPU fellowship, AP2009-3854, from the Spanish Ministry of Education.

Contents

Chapter 1

Introduction

The Internet has become one of the most significant changes experienced by the world in the last decades. It allows us to share information, socialize and buy and sell goods faster, cheaper and more efficiently than ever before. Furthermore, it simplifies the distribution of contents and information to a large number of users.

There are several settings where some control on the distribution process is required, in terms of disallowing users from redistributing their own copy of the content freely. These settings are typically motivated by the kind of content distributed including, but not limited to, personal documents, industrial secrets, classified information and copyrighted material.

One may attempt to tackle the redistribution problem by implementing a *copy prevention* mechanism, i.e., implementing techniques that impede users from making copies of the received content. However, it is generally accepted by numerous experts in the field that it is theoretically impossible to completely prevent users from making and distributing copies of the content that they have received. The main argument for this assertion is the fact that any kind of content needs to be "read" somehow to be used. Hence, a user could simply implement a reader that it first reads the content, and then it writes an exact copy of what was read.

Another alternative consists in discouraging users from redistributing their copy of the content, rather than trying to avoid this from happening. This is achieved by *copy detection* techniques. By using these techniques, users are free to redistribute

their own copy. Nevertheless, the distributor reserves the right to prosecute and/or penalize in some way users found guilty of an illegitimate redistribution. Obviously, if there is only a single user in the system, then it becomes trivial to identify the guilty party when the content appears published elsewhere. Problems arise when the content has been distributed to multiple legitimate users. If all of them have received the same exact object, then it becomes impossible to identify dishonest members.

Therefore, it is clear that a distributor implementing a copy detection technique must deliver a unique object to every authorized user. Each copy of the content can be made unique by embedding in it a mark that identifies each user. By making each copy unique, plain redistribution is ruled out. However a group of *traitors* (dishonest users) could create a *pirated copy*, which is a "combination" of their copies of the content, and distribute this new copy. We call such an attack a *collusion attack*. The precise way in which the traitors combine their copies to generate a pirated copy will be made precise below. The goal of the pirated copy is to disguise the identity of the colluders once it is redistributed. What is worse, it could be the case that the pirated copy be very similar to the copy of an innocent user, what could lead the distributor to accuse that user incorrectly. Therefore, the distributor faces the problem of identifying the real traitors using the information contained in the pirated copy.

The original idea of making copies unique by embedding a different marks for each user was introduced in [11]. There, this technique was coined under the name of *fingerprinting*, by analogy to human fingerprints, and was subsequently adopted by many authors, e.g. [11, 12, 13, 14, 15, 16, 17, 18]; see also the tutorial paper [19]. Hence, it is common to call the individual marks *fingerprints*, and the set of all user marks a *fingerprinting code*.

1.1 General Considerations

A robust fingerprinting scheme should be designed so that innocent users are never incorrectly accused. Also, it should allow the identification of traitors that have

participated in a collusion attack. These two objectives are usually difficult to achieve, and it is common to allow some (small) error probability for these events.

Now, let us describe a common set of considerations that are usually assumed in a fingerprinting scheme.

First, the distributor chooses a set of marks, which constitutes the fingerprinting code, where each mark identifies one of the possible users of the system. Next, a subset of redundant positions in the content is selected, and the mark is embedded in these positions. The marked copies are delivered correspondingly to the users. This set of redundant positions are constant for all the copies. Regarding the embedding process, it must satisfy some properties. On one hand, the marked content must not differ substantially from the original content and must retain the same functionality. On the other hand, the users should not be able to remove or degrade the mark once it is embedded without rendering the content unusable.

As customary in the literature, we will assume a setting coined by Boneh and Shaw [13, 14] as the *marking assumption*. In this setting, a coalition of traitors may attempt to discover the positions of the fingerprints by comparing their copies, which will reveal a number of differences, at the positions where their fingerprints differ. Now, they generate a pirated copy following the assumption that the positions where the traitors have not found any difference must remain unchanged in the pirated copy. This is assumed because the traitors do not have any information about what positions in the content are redundant, and modifying arbitrary positions may damage the content. In the positions where they have found a difference, they are allowed to change them in some way, possibly making that position unreadable.

Once an illegitimate redistribution has been found, the distributor extracts the embedded mark. Using this information, the goal is to identify at least one of the colluding members. Therefore, the set of user marks that constitute the fingerprinting code must have *tracing properties*. Identifying users from arbitrary-size collusion attacks is a very ambitious and restrictive requirement that imposes strong constraints on the design of the fingerprinting code. Therefore, this requirement is usually relaxed by bounding the maximum size of the coalitions to a certain number of traitors.

For the reasons given above, when studying a fingerprinting scheme, it becomes sufficient to study the set of marks, i.e., the fingerprinting code. This is because the positions where the mark has not been embedded will be identical in all the copies of the content, and according to the marking assumption, they will be unaltered in the pirated copy, providing no information about the traitors.

1.2 Motivation and Objectives

The main objective of this dissertation is the study and design of codes appropriate for fingerprinting settings under the presence of collusion attacks. We will be mainly concerned with the existence conditions of such codes, and also with the design of explicit constructions. Furthermore, it will also be a paramount topic in our discussion the design of efficient identification algorithms, i.e., in polynomial time in the code length.

It is a well-known result that conventional error-correcting codes over a sufficiently large alphabet and with a sufficiently large minimum distance posses the desired identification capabilities. In other words, there exist solutions to the fingerprinting problem that allow the distributor identify traitors with zero-error probability. However, relying only in the use of conventional error-correcting codes have two drawbacks. On one hand, the use of these codes assume that the traitors generate pirated copies in a very restricted way, which may be a very optimistic supposition. On the other hand, the use of large alphabets is difficult to handle for the marking-insertion layer. To overcome these problems, the idea of code concatenation has been used. Hence, the study of how code concatenation can enable us to obtain new families of fingerprinting codes with small error, and how identification can be done in polynomial time, also constitute objectives of the dissertation.

Another objective of the dissertation is the study of the combinatorial properties of codes used in fingerprinting schemes. Codes with separating and traceability properties have proved to be useful in these schemes. We will explore how these codes, or modified versions of these codes, can be used to construct fingerprinting codes.

Finally, we remark that in the present work we will not be concerned about the nontrivial process of embedding and extracting marks from the content. Rather, we will focus on how to design a set of marks that allow the distributor identify traitors under the presence of collusion attacks.

1.3 Contributions

Below we list the publications derived from this work, in order of appearance.

J. Moreira, M. Fernández, and M. Soriano, "A note on the equivalence of the traceability properties of Reed-Solomon codes for certain coalition sizes," in *Proc. IEEE Int. Workshop Inform. Forensics, Security (WIFS)*, London, United Kingdom, Dec. 2009, pp. 36–40

J. Moreira, M. Fernández, and M. Soriano, "Propiedades de trazabilidad de los códigos de Reed-Solomon para ciertos tamaños de coalición," in *Proc. Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, Tarragona, Spain, Sep. 2010, pp. 413–417

M. Fernández, J. Moreira, and M. Soriano, "Identifying traitors using the Koetter-Vardy algorithm," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 692–704, Feb. 2011

M. Fernández, G. Kabatiansky, and J. Moreira, "Almost separating and almost secure frameproof codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Saint Petersburg, Russia, Aug. 2011, pp. 2696–2700

J. Moreira, G. Kabatiansky, and M. Fernández, "Lower bounds on almost-separating binary codes," in *Proc. IEEE Int. Workshop Inform. Forensics, Security (WIFS)*, Foz do Iguaçu, Brazil, Nov. 2011, pp. 1–6

M. Á. Simarro-Haro, J. Moreira, M. Fernández, M. Soriano, A. González, and F. J. Martínez-Zaldívar, "Parallelization of the interpolation process in the Koetter-Vardy soft-decision list decoding algorithm," in *Proc. Int. Conf. Comput. Math. Methods (CMMSE)*, La Manga, Spain, Jul. 2012, pp. 1102–1110

M. Á. Simarro-Haro, J. Moreira, M. Fernández, M. Soriano, A. González, and F. J. Martínez-Zaldívar, "Paralelización en la interpolación de la decodificación por listas de códigos Reed-Solomon," in *Proc. Jornadas de Paralelismo (JP)*, Elx, Spain, Sep. 2012

J. Moreira, M. Fernández, and M. Soriano, "On the relationship between the traceability properties of Reed-Solomon codes," *Adv. Math. Commun.*, vol. 6, no. 4, pp. 467–478, Nov. 2012

J. Moreira, M. Fernández, and G. Kabatiansky, "Fingerprinting basado en códigos cuasi separables con identificación eficiente," in *Proc. Jornadas de Ingeniería Telemática (JITEL)*, Granada, Spain, Oct. 2013 (To appear)

J. Moreira, M. Fernández, and G. Kabatiansky, "Constructions of almost secure-frameproof codes based on small-bias probability spaces," in *Proc. Int. Workshop Security (IWSEC)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 8231, Okinawa, Japan, Nov. 2013, pp. 53–67

1.4 Notation and Conventions

Here we list the most relevant notation used.

$A = \{a_1, \ldots, a_n\}$	set having elements $a_1, \ldots, a_n$
$\|A\|$	size of the set A
$\emptyset$	empty set
A^n	nth cartesian power of the set A
$A \setminus B$	set difference
r.v.	random variable
pmf	probability mass function
$E[\,]$, $E_f[\,]$	expectation / expectation over the pmf f
$H_2(x)$	binary entropy function, $H_2(x) = -x \log_2 x - (1-x)\log_2(1-x)$

$D(x\|\|y)$	Kullback-Leibler divergence between two Bernoulli distributed r.v.'s of parameters x and y, respectively, $D(x\|\|y) = x\log_2(x/y) + (1-x)\log_2((1-x)/(1-y))$
$q!$	factorial, $q! = q(q-1)\cdots 1$
$q^{\underline{j}}$	falling factorial, $q^{\underline{j}} = q(q-1)\cdots(q-j+1)$
$\binom{n}{k}$	binomial coefficient, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
$\left\{{n \atop k}\right\}$	Stirling number of the second kind, $\left\{{n \atop k}\right\} = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$
Q	finite alphabet, i.e., a nonempty finite set
$\mathbb{F}_q$	finite field of q elements
$\mathbf{u} = (u_1, \ldots, u_n)$	vector with entries over an alphabet Q, see p. 9
$d(\mathbf{u}, \mathbf{v})$	Hamming distance between vectors $\mathbf{u}$ and $\mathbf{v}$, see p. 9
$\mathrm{s}(\mathbf{u}, \mathbf{v})$	similitude between vectors $\mathbf{u}$ and $\mathbf{v}$, $\mathrm{s}(\mathbf{u}, \mathbf{v}) = n - d(\mathbf{u}, \mathbf{v})$, see p. 9
$\mathrm{s}(\mathbf{u}, \boldsymbol{\mathcal{Z}})$	similitude between vector $\mathbf{u}$ and the "set vector" $\boldsymbol{\mathcal{Z}}$, see p. 9
C	a code
$d(C)$	minimum distance of the code C, see p. 10
$R(C)$	rate of the code C, see p. 10
(n, M)-code	a code (over an alphabet Q) of length n and size M, i.e., a subset of Q^n of size M, see p. 10
$[n, k]$-code	a linear code (over a finite field $\mathbb{F}_q$) of length n and dimension k, i.e., a vector subspace of $\mathbb{F}_q^n$ of dimension k, see p. 10
c-coalition	a subset of size c of a code, see p. 10
U, V	c-coalitions
$P_i(U)$	projection of the c-coalition $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\}$ on the ith position, $P_i(U) = \{u_i^1, \ldots, u_i^c\}$, see p. 10
$\mathrm{desc}(U)$	narrow-sense envelope of U, see Definition 2.1, p. 10
$\mathrm{desc}^*(U)$	expanded narrow-sense envelope of U, see Definition 2.2, p. 11
$\mathrm{Desc}(U)$	wide-sense envelope of U, see Definition 2.2, p. 11
$\mathrm{Desc}^*(U)$	expanded wide-sense envelope of U, see Definition 2.2, p. 11
$\mathcal{E}(U)$	an arbitrary envelope of U, see p. 11
IPP, IPP code	identifiable parent property, code with the identifiable parent property, see Definition 2.8, p. 14

TA, TA code	traceability property, code with the traceability property, see Definition 2.9, p. 15
$R(\mathcal{C})$	rate of a family of codes $\mathcal{C} = \{C_t\}_{t\in T}$, see Definition 2.11, p. 17
$R(\mathcal{C})$	asymptotical rate of a sequence of codes $\mathcal{C} = (C_i)_{i\geq 1}$, see p. 60
$R_q^{\text{fing}}(c)$	maximal asymptotically achievable c-fingerprinting rate, see p. 20
$R_q^{\text{sep}}(n, c, c')$	maximal rate of a q-ary (c, c')-separating code of length n, see p. 57
$R_q^{\text{sep}^*}(c)$	maximal asymptotical rate among all asymptotically almost (c, c)-separating families, see p. 60
$R_q^{\text{SFP}^*}(c)$	maximal asymptotical rate among all asymptotically almost c-secure frameproof families, see p. 67
$\deg f(x)$	degree of the polynomial $f(x)$
$\operatorname{im} f$	image of the application f
$\ker f$	kernel of the application f
$v(j; q, c)$, $v(j)$	pmf, evaluated at j, of an r.v. that counts the number of different symbols of a q-ary vector of length c chosen uniformly at random, see Lemma 4.3, p. 58
$p_{q,c,c'}^{\text{disj.}}$	probability that two q-ary vectors of lengths c and c', respectively, chosen uniformly and independently at random, have no common element, see Lemma 4.4, p. 58
$h(k; N, K, n)$	pmf, evaluated at k, of a hypergeometric r.v. with a total size of the population N, number of items with the desired characteristic K, and number of samples drawn n, see p. 65
$N(j; U)$	number of positions where the elements of the c-coalition U have j different symbols, see p. 61
$Z(x; U)$	number of positions where all the elements of the c-coalition U have the symbol x, see p. 63
$\nu_S(\mathbf{a}; A)$	number of rows of a binary array (binary matrix) A whose projection onto the indices of the subset S equals the vector $\mathbf{a} \in \mathbb{F}_2^s$, see p. 83
$\theta(U, V)$	number of separating positions between U and V, see p. 98
$\theta_{c,c'}(C)$, $\theta_{c,c'}$	for a code C, minimum value of $\theta(U, V)$ for disjoint $U, V \subseteq C$ such that $\lvert U\rvert = c$ and $\lvert V\rvert = c'$, see p. 98

Chapter 2

Preliminaries and Background

In this chapter we present some basic elements of coding theory and fingerprinting that will be used throughout the dissertation. This, in turn, will allow us to introduce some notation and conventions.

Let $q \geq 2$ be an integer. A q-ary *alphabet* Q is a nonempty set of size q. For any integer $n \geq 1$, let Q^n denote the set of all possible n-tuples over the alphabet Q. We denote the elements of Q^n in boldface, e.g. $\mathbf{u} = (u_1, \ldots, u_n) \in Q^n$. The *(Hamming) distance* between two elements $\mathbf{u}, \mathbf{v} \in Q^n$ is denoted $d(\mathbf{u}, \mathbf{v})$, and is defined as the number of positions $1 \leq i \leq n$ where $\mathbf{u}$ and $\mathbf{v}$ differ,

$$d(\mathbf{u}, \mathbf{v}) \stackrel{\text{def}}{=} |\{i : u_i \neq v_i, 1 \leq i \leq n\}|.$$

Sometimes, it will also be convenient to talk about the *similitude* of $\mathbf{u}$ and $\mathbf{v}$, denoted $\mathrm{s}(\mathbf{u}, \mathbf{v})$, and defined as

$$\mathrm{s}(\mathbf{u}, \mathbf{v}) \stackrel{\text{def}}{=} |\{i : u_i = v_i, 1 \leq i \leq n\}| = n - d(\mathbf{u}, \mathbf{w}).$$

Moreover, for a set of subsets of the alphabet, $\mathcal{Z}_i \subseteq Q$, for $1 \leq i \leq n$, and an element $\mathbf{u} = (u_1, \ldots, u_n) \in Q^n$, we define the similitude between $\mathbf{u}$ and the "set vector" $\boldsymbol{\mathcal{Z}} = (\mathcal{Z}_1, \ldots, \mathcal{Z}_n)$ as

$$\mathrm{s}(\mathbf{u}, \boldsymbol{\mathcal{Z}}) \stackrel{\text{def}}{=} |\{i : u_i \in \mathcal{Z}_i, 1 \leq i \leq n\}|. \tag{2.1}$$

Let Q be a q-ary alphabet. An (n, M)-*code* C over Q is a subset of Q^n of size M. The parameter n is called the *length* of the code. If Q is the finite field of q elements, we denote it by $\mathbb{F}_q$. A code C is a *linear* $[n, k]$-*code* over $\mathbb{F}_q$ if $C \subseteq \mathbb{F}_q^n$ is a vector subspace of dimension k. The elements of a code are called *codewords*, and the matrix formed with the codewords as rows is called *codebook*. The *minimum distance* of a code C, denoted $d(C)$, is the smallest Hamming distance between any two of its codewords,

$$d(C) \stackrel{\text{def}}{=} \min_{\mathbf{u},\mathbf{v}\in C}\{d(\mathbf{u}, \mathbf{v}) : \mathbf{u} \neq \mathbf{v}\}.$$

The *rate* of a q-ary (n, M)-code C, denoted $R(C)$, is defined as

$$R(C) \stackrel{\text{def}}{=} n^{-1} \log_q M.$$

Obviously, if C is a linear $[n, k]$-code, we have $R(C) = k/n$.

For a code C, we call a subset of codewords $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C$ of size c a *c-subset* or *c-coalition*. Given a c-coalition U, we denote by $P_i(U)$ the *projection* of U on the ith position, i.e., the set of elements of the code alphabet at the ith position,

$$P_i(U) \stackrel{\text{def}}{=} \{u_i^1, \ldots, u_i^c\}. \tag{2.2}$$

A position i is *undetectable* for coalition U if the codewords of U match in their ith position, i.e., $u_i^1 = \cdots = u_i^c$, or equivalently, $|P_i(U)| = 1$. A position that does not satisfy this property is called *detectable*.

According to the marking assumption [13, 14], introduced in Section 1.1, when a c-coalition U generates a forged copy of the content, the undetectable positions remain unchanged in the pirated word. For the detectable positions, the traitors are allowed to alter them in some way, possibly making them unreadable. This is a very natural approach to model the generation of a pirated word, since when a group of up to c traitors generates a pirated content, a comparison of their copies will only reveal the detectable positions. How the traitors set the detectable positions in the pirated words gives rise to different fingerprinting settings, leading to different results.

Definition 2.1. Let C be an (n, M)-code over an alphabet Q, and consider the c-coalition $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C$. We say that $\mathbf{z} \in Q^n$ is a *descendant* of coalition U if for each position $1 \leq i \leq n$ there exists a $\mathbf{u} \in U$ such that $z_i = u_i$. We call $\mathbf{u}$ a *parent* of $\mathbf{z}$. The set of all the descendants of U is denoted $\mathrm{desc}(U)$,

$$\mathrm{desc}(U) \stackrel{\mathrm{def}}{=} \{\mathbf{z} \in Q^n : z_i \in P_i(U)\}. \tag{2.3}$$

Also, the *c-descendant code* of C, denoted $\mathrm{desc}_c(C)$, is defined as

$$\mathrm{desc}_c(C) \stackrel{\mathrm{def}}{=} \bigcup_{U \subseteq C, |U| \leq c} \mathrm{desc}(U). \tag{2.4}$$

The set of descendants that we have just introduced is also known as the *narrow-sense envelope* by some authors [15]. This definition can be extended for some other natural settings as follows.

Definition 2.2. Let C be an (n, M)-code over an alphabet Q, and consider the c-coalition $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C$. Also, let '$*$' denote any element, $* \notin Q$. Then, we define

1) the *expanded narrow-sense envelope* of U, denoted $\mathrm{desc}^*(U)$, as the set of all words $\mathbf{z} \in (Q \cup \{*\})^n$ such that for all undetectable positions i we have $z_i \in P_i(U)$, and for all detectable positions j we have $z_j \in P_j(U) \cup \{*\}$;
2) the *wide-sense envelope* of U, denoted $\mathrm{Desc}(U)$, as the set of all words $\mathbf{z} \in Q^n$ such that for all undetectable positions i we have $z_i \in P_i(U)$, and for all detectable positions j we have $z_j \in Q$;
3) the *expanded wide-sense envelope* of U, denoted $\mathrm{Desc}^*(U)$, as the set of all words $\mathbf{z} \in (Q \cup \{*\})^n$ such that for all undetectable positions i we have $z_i \in P_i(U)$, and for all detectable positions j we have $z_j \in Q \cup \{*\}$.

Similarly as in (2.4), denote the corresponding descendant codes, for coalitions of size at most c, by $\mathrm{desc}^*_c(C)$, $\mathrm{Desc}_c(C)$ and $\mathrm{Desc}^*_c(C)$, respectively.

The symbol '$*$' above denotes an erased or unreadable position in the pirated word. For a code C and a c-coalition $U \subseteq C$, we denote by $\mathcal{E}(U)$ an arbitrary envelope of those defined above, and $\mathcal{E}_c(C)$ the corresponding c-descendant code. Hence $\mathcal{E}(U)$

can be interpreted as the set of pirated words that coalition U is able to generate in a given fingerprinting setting. Note that $U \subseteq \mathcal{E}(U)$, and also note that the marking assumption is fulfilled in every definition of the envelope model given above.

Example 2.3. Let $U = \{\mathbf{u}^1, \mathbf{u}^2\} \in Q^5$, with $Q = \{1, 2, 3, 4\}$, $\mathbf{u}^1 = (3, 4, 1, 2, 3)$ and $\mathbf{u}^2 = (4, 2, 1, 3, 3)$, then

- $\text{desc}(U) = \{3, 4\} \times \{2, 4\} \times \{1\} \times \{2, 3\} \times \{3\}$,
- $\text{desc}^*(U) = \{3, 4, *\} \times \{2, 4, *\} \times \{1\} \times \{2, 3, *\} \times \{3\}$,
- $\text{Desc}(U) = Q \times Q \times \{1\} \times Q \times \{3\}$,
- $\text{Desc}^*(U) = (Q \cup \{*\}) \times (Q \cup \{*\}) \times \{1\} \times (Q \cup \{*\}) \times \{3\}$.

When an illegal redistribution of the content has occurred, the goal of the distributor is to identify at least one of the c traitors from the coalition $U \subseteq C$ that generated the pirated content, using the pirated word $\mathbf{z} \in \mathcal{E}(U)$ observed in it. Recall that all the undetectable positions are common to all the traitor codewords and to the pirated word. Also, some of the detectable positions *may* coincide with some traitor codeword. Using this information the distributor tries to perform the identification using a *decoding* or *identification algorithm*, which can be regarded as a function A

$$A : \mathcal{E}_c(C) \to C \cup \{?\},$$

where '?' denotes an unknown element. An identification error occurs when we have $A(\mathbf{z}) \notin U$. Observe, however, that two types of error can be considered. On one hand, the *completeness error* (false negative), when $A(\mathbf{z}) =?$, and on the other hand, the *soundness error* (false positive), when $A(\mathbf{z}) \in C \setminus U$. Obviously, the latter type is far more severe than the former, since the distributor would be accusing an innocent user.

Remark 2.4. There are a variety of fingerprinting codes where the output of the identification algorithm need not be a single codeword, but a subset of codewords from the code C,

$$A : \mathcal{E}_c(C) \to 2^C.$$

This fact depends on the nature of the codes and its identification algorithm. In this case, the completeness error occurs when $A(\mathbf{z}) = \emptyset$, and the soundness error when $A(\mathbf{z}) \cap (C \setminus U) \neq \emptyset$. See [13, 14, 18] for examples of fingerprinting codes with identification algorithms that may produce more than one output codeword.

2.1 Zero-Error Fingerprinting

Informally, we talk about *zero-error fingerprinting* when the distributor has mechanisms to identify unambiguously a traitor of any coalition of size at most c. In other words, there exists a code C and an identification algorithm A such that for any c-coalition $U \subseteq C$ and any $\mathbf{z} \in \mathcal{E}(U)$ we have $A(\mathbf{z}) \in U$. This is, indeed, the ideal situation in a fingerprinting setting. Note that it cannot be guaranteed that the distributor finds more than one colluding user, since the remaining traitors could be passive in the generation of the pirated content, or may contribute with few symbols to the pirated word.

Definition 2.5 ([13, 14])**.** A code C is *totally c-secure* if for any c-coalition U there is an identification algorithm A such that $A(\mathbf{z}) \in U$ for any $\mathbf{z} \in \mathcal{E}(C)$.

Sadly, there are no totally c-secure codes when $\mathcal{E}(C)$ is an envelope model different from the narrow-sense one defined above, and hence, zero-error probability cannot be guaranteed in the identification process.

Theorem 2.6 ([13, 14])**.** For $q \geq 2, c \geq 2$ and $M \geq 3$ there are no totally q-ary c-secure (n, M)-codes under the wide-sense and the expanded envelope models.

In this section we will restrict our discussion to the particular case of the narrow-sense envelope model. Let us introduce some codes that have interesting properties for a fingerprinting setting under this model.

Definition 2.7. A code C is (c, c')-*separating* if for every pair of disjoint subsets $U, V \subseteq C$ with $|U| = c$ and $|V| = c'$ there is a position $1 \leq i \leq n$ such that their projections on that position have empty intersection, i.e.,

$$P_i(U) \cap P_i(V) = \emptyset.$$

Clearly, a code that is (c, c')-separating is also (t, t')-separating, for $t \leq c$ and $t' \leq c'$.

The separating property was first discussed in [20], and has been investigated by many authors [21, 22, 23, 24, 25, 26]. Recently, more attention has been paid to separating codes in connection with fingerprinting settings. In the crypto literature, $(c, 1)$- and (c, c)-separating codes are also known as *c-frameproof codes* and *c-secure frameproof codes*, respectively, [13, 14, 27, 28].

The connection between separating and fingerprinting codes is straightforward. Assume that a fingerprinting code C has the $(c, 1)$-separating property. Then, no coalition of size $\leq c$ will be able to generate a pirated word $\mathbf{z} \in \mathrm{desc}_c(C)$ that coincides with the fingerprint of an innocent user. Moreover, using a (c, c)-separating code, a given coalition can not even claim that the pirated word was generated by a disjoint coalition of size $\leq c$, since for disjoint coalitions $U, V \in C$ it is easy to see that

$$\mathrm{desc}(U) \cap \mathrm{desc}(V) = \emptyset.$$

Still, the separating property is only a necessary condition to achieve unambiguous identification of traitors. To see that it is not sufficient, consider the case $c = 2$ and the code

$$C = \{\mathbf{u}^1 = (0, 0, 0), \mathbf{u}^2 = (0, 1, 1), \mathbf{u}^3 = (1, 1, 0), \mathbf{u}^4 = (1, 0, 1)\},$$

which is $(2, 2)$-separating. Since

$$(0, 1, 0) \in \mathrm{desc}(\{\mathbf{u}^1, \mathbf{u}^2\}) \cap \mathrm{desc}(\{\mathbf{u}^1, \mathbf{u}^3\}) \cap \mathrm{desc}(\{\mathbf{u}^2, \mathbf{u}^3\}),$$

one cannot decide which of the three possible pairs of codewords is the actual coalition of traitors that generated the pirated word $(0, 1, 0)$.

Now, we present codes with sufficient conditions to allow identification with zero-error under the narrow-sense envelope model.

Definition 2.8. A code $C \subseteq Q^n$ has the *c-identifiable parent property* (c-IPP) if for all $\mathbf{z} \in Q^n$, either $\mathbf{z} \notin \mathrm{desc}_c(C)$ or

$$\bigcap_{\substack{U \subseteq C, |U| \leq c \\ \text{s.t. } \mathbf{z} \in \mathrm{desc}(U)}} U \neq \emptyset. \tag{2.5}$$

Note that for a c-IPP code the intersection of all coalitions of size $\leq c$ that can generate a given pirated word is nonempty. In particular, the codewords that lie the intersection (2.5) belong to the coalition that generated the pirated word and can be accused as traitors. This means that the distributor could simply apply an identification algorithm A that consists in finding the intersection of all possible c-coalitions. Hence, for a code of size M, the identification process runs in time $O\binom{M}{c}$ in the general case.

Definition 2.9. A code C has the *c-traceability property* (c-TA) if for all subsets $U \subseteq C$ of size at most c, if $\mathbf{z} \in \mathrm{desc}(U)$, then there exists a $\mathbf{u} \in U$ such that $d(\mathbf{z}, \mathbf{u}) < d(\mathbf{z}, \mathbf{w})$ for all $\mathbf{w} \in C \setminus U$.

That is, in a c-TA code the closest codeword to a descendant of a c-coalition U, in terms of Hamming distance, is in U.

It is easy to see that every TA code is an IPP code [28, 29, 30]. The main benefit of using TA codes is that the identification process runs in time $O(M)$. Nevertheless the TA property imposes more restrictions to the code than the IPP property; see for example [28].

The concepts of IPP and TA codes were originated in [29] (later in [30]). However, no specific name was given to such codes. IPP codes where further studied in [31]. There, the authors coined the term "IPP," that has been widely adopted in the crypto literature. Also, IPP and TA codes have been investigated in [28] under the names presented here.

A simple, sufficient condition for an (n, M)-code C to posses the TA property was first presented in [29, 30]. Namely, if

$$d(C) > (1 - 1/c^2)n, \tag{2.6}$$

the code is c-TA. In addition, the following chain of implications are also well-known results:

$$d(C) > (1 - 1/c^2)n \Rightarrow c\text{-TA} \Rightarrow c\text{-IPP} \Rightarrow (c, c)\text{-separating} \Rightarrow (c, 1)\text{-separating}. \quad (2.7)$$

These results were presented later in the form of a theorem in [28]. Moreover, it is not difficult to see that

$$d(C) > (1 - 1/c)n \Rightarrow (c, 1)\text{-separating}.$$

At a first glance, IPP codes seem to be the solution to the fingerprinting problem in the narrow-sense envelope model. However, an important limitation of IPP codes is that the size of the alphabet severely limits their collusion-resistant properties, as stated in the following lemma.

Lemma 2.10 ([28])**.** Suppose C is a q-ary (n, M)-code and $n - 1 \geq c \geq q - 1$. Then C is not a c-IPP code.

The algorithms used to embed marks in content have a worse performance as the size of the code alphabet grows. Also, since we are mainly interested in the distribution of digital contents, the case of binary alphabets is of high interest. However the previous lemma sadly states that there are no c-IPP (or c-TA) codes over binary alphabets.

Finally, observe that the code $C = \{(1, \ldots, 1), \ldots, (q, \ldots, q)\}$ over the alphabet $Q = \{1, \ldots, q\}$ is a trivial c-IPP code regardless of its length. A c-IPP code of size M over a q-ary alphabet is nontrivial if $M > \max\{c, q\}$.

2.2 Nonzero-Error Fingerprinting

The use of IPP codes poses severe limitations in a fingerprinting setting: the size of the code alphabet is lower bounded by the coalition size, and the setting is restricted to the narrow-sense envelope model. These two limitations can be overcome if we

allow some error in the identification process. Moreover, a single deterministic code is not enough to achieve arbitrarily small decline in the identification error [13, 14], and some "randomness" over a family of (fingerprinting) codes is required.

Definition 2.11. A *family of q-ary* (n, M)*-codes* is an indexed set, denoted $\mathcal{C} = \{C_t\}_{t\in T}$, where each C_t is a q-ary (n, M)-code, and T is a finite set of elements called *keys*. The rate of the family is defined as

$$R(\mathcal{C}) \stackrel{\text{def}}{=} n^{-1} \log_q M.$$

To use a family of codes $\mathcal{C} = \{C_t\}_{t\in T}$ the distributor chooses a code $C_t \in \mathcal{C}$ with probability $\pi(t)$. The family $\mathcal{C}$ and the pmf π are publicly known, but the specific code C_t used by the distributor is kept secret. It is usual to assume that $\pi(t) = |T|^{-1}$ for all $t \in T$, and unless otherwise stated in this work, this will be the default probability distribution in the analysis of the families of fingerprinting codes constructed. The result of this random experiment is sometimes called a *randomized code*. Now, the fingerprints assigned to the users correspond to the codewords of the selected code. By an abuse of language, often is called "an identification algorithm for the family of codes" what is in fact a collection of identification algorithms $\mathcal{A} = \{A_t\}_{t\in T}$,

$$A_t : \mathcal{E}_c(C_t) \to C_t \cup \{?\}, \quad t \in T.$$

Upon receiving a pirated word the distributor selects the appropriate algorithm A_t to perform the identification of traitors.

Let $\mathcal{C} = \{C_t\}_{t\in T}$ be a family of codes and let π be a pmf on T. Moreover, let $\mathcal{A} = \{A_t\}_{t\in T}$ be the corresponding set of identification algorithms. Since each code C_t has size M, this is the maximum number of users that the distributor can allocate. Let us number these users arbitrarily from 1 to M, and let $C_t(i)$ denote the corresponding codeword from C_t assigned by the distributor to the ith user. Similarly, for a group of users of indices $X \subseteq \{1, \ldots, M\}$, let $C_t(X)$ denote the set of assigned codewords from C_t. If code C_t has been chosen and the pirated word $\mathbf{z}$ is observed, then the distributor will accuse the user corresponding to codeword $A_t(\mathbf{z})$.

Let X be a group of, at most, c users and let their set of codewords be $C_t(X)$. Following the discussion from [15], their strategy can be modeled, in the most general way, by the pmf

$$f_X(\mathbf{z}|C_t(X)) \stackrel{\text{def}}{=} \Pr\{\text{coalition of users } X \text{ produces pirated word } \mathbf{z}, \text{ given that they observe codewords } C_t(X)\}.$$

Obviously, if $\mathcal{E}$ is the envelope model under consideration, we have $f_X(\mathbf{z}|C_t(X)) = 0$ when $\mathbf{z} \notin \mathcal{E}(C_t(X))$.

On the other hand, the most general decision rule for the distributor can be modeled with the pmf

$$f_{\text{dist}}(i|\mathbf{z};t) \stackrel{\text{def}}{=} \Pr\{C_t(A_t(\mathbf{z})) = i\},$$

that is, the probability that the ith user is returned by the identification algorithm A_t given that code C_t was chosen and pirated word $\mathbf{z}$ was observed.

Then, the error probability of the distributor in identifying a member of X under this envelope model can be expressed as

$$p_e(X, f_X) \stackrel{\text{def}}{=} E_\pi\Big[\sum_{i \notin X} \sum_{\mathbf{z} \in \mathcal{E}(C_t(X))} f_{\text{dist}}(i|\mathbf{z};t) \cdot f_X(\mathbf{z}|C_t(X))\Big],$$

where $E_\pi[\ ]$ denotes the expectation with respect the pmf π.

It is a natural assumption that the coalition of users X is interested in designing a strategy f_X such that this error probability is maximized. This fact motivates the following definition.

Definition 2.12. Let $\mathcal{C} = \{C_t\}_{t \in T}$ be a family of (n, M)-codes with pmf π on T, equipped with a set of identification algorithms $\mathcal{A} = \{A_t\}_{t \in T}$ modeled by the decision rules with pmf f_{dist}. We say that the family $\mathcal{C}$ is *c-secure with ε-error* under envelope model $\mathcal{E}$ if

$$p_e(\mathcal{E}; \mathcal{C}, T, \pi, f_{\text{dist}}) \stackrel{\text{def}}{=} \max_{X:|X| \leq c} \max_{f_X} p_e(X, f_X) \leq \varepsilon. \tag{2.8}$$

Note that the error probability (2.8) depends on the choice of the family $\mathcal{C}$, the pmf's f_{dist} and π, and, if the number of keys (codes in the family $\mathcal{C}$) is unrestricted, also from the set T. It is assumed that the distributor chooses these parameters in order to minimize this error probability. Hence, for a given envelope model $\mathcal{E}$, we define

$$p_e(\mathcal{E}; n, M, c) \stackrel{\text{def}}{=} \min_{\mathcal{C},T,\pi,f_{\text{dist}}} p_e(\mathcal{E}; \mathcal{C}, T, \pi, f_{\text{dist}}), \tag{2.9}$$

where the minimization is understood over all families of q-ary (n, M)-codes and collusion attacks carried out by groups of $\leq c$ traitors.

The general fingerprinting problem consists in finding $p_e(\mathcal{E}; n, M, c)$, codes and identification algorithms achieving this error probability. For a more detailed exposition, see [15].

Some important conclusions about the fingerprinting problem are also discussed in [15]. Even though in the design of a fingerprinting scheme the associated error probability (2.9) highly depends on the envelope model that is considered, some equivalences are identified. For example, for binary codes, the error probability takes the same value, regardless of the specific envelope model considered. On the other hand, for q-ary codes, (2.9) coincides in the wide-sense and expanded wide-sense envelope models. Moreover, for a desired error probability ε, it is also shown that

$$|T| \geq \frac{1}{(c+1)\varepsilon}.$$

This means that for a family of codes to achieve exponential decline of the error probability ε, the number of codes in the family must grow exponentially with the code length.

2.2.1 Optimal Codes

In practical settings, one of the main concerns of the distributor is to obtain families of fingerprinting codes of maximal rate, i.e., families of codes that allocate the maximum number of users for a given code length n and error probability ε. Equivalently, the problem can be restated to obtaining families of codes with the shortest possible length

for a given number of users to allocate M and error probability ε. In connection with this problem, a figure of high theoretical value is the asymptotical rate of a family of codes.

Let $(\mathcal{C}_i)_{i\geq 1}$ be a sequence of families of q-ary codes of growing length n_i, where each family $\mathcal{C}_i$ is c-secure with ε_i-error. We say that R is an *asymptotically achievable c-fingerprinting rate* if there exists such a sequence with

$$\lim_{i\to\infty} \varepsilon_i = 0, \qquad \text{and} \qquad \liminf_{i\to\infty} R(\mathcal{C}_i) = R.$$

We denote by $R_q^{\text{fing}}(c)$ the maximal possible asymptotically achievable c-fingerprinting rate among all the sequences of families of q-ary codes. This figure is sometimes called the *q-ary c-fingerprinting capacity*.

Some important values of the q-ary c-fingerprinting capacity are known. For example, from [32] we have $0.25 \leq R_2^{\text{fing}}(2) \leq 0.322$ and $0.083 \leq R_2^{\text{fing}}(3) \leq 0.199$. Also, from [18, 32], we have $O(1/t^2) \leq R_q^{\text{fing}}(c) \leq O(1/t)$.

2.3 Code Concatenation

Frequently, codes with the IPP or the TA property are used as outer codes in concatenated fingerprinting constructions [13, 14, 15], and in traitor-tracing schemes [29, 30]. This connection was also pointed out in the seminal paper by Boneh and Shaw [13,14]. As an example, we can consider pay-TV systems, where each authorized user is given a set of keys that allows him to decrypt the content. These keys are usually contained in a decoder box. This particular case is clearly restricted to the narrow-sense envelope model, since in a collusion attack, the traitors combine some of their keys to construct a pirated decoder. In each position, the pirated word must contain one of the colluding traitors keys, otherwise, the pirated receiver will not be capable of decoding the TV signal. Again, when a pirated decoder is found, a traitor-tracing scheme allows the distributor to identify at least one of the guilty users, using the keys inside the decoder.

As said above, in the binary case the identification process will always be subject to a certain error probability. To construct practical binary fingerprinting codes many authors [13, 14, 15, 33] have used the idea of code concatenation [34].

Construction 2.13. Let C_{out} be an (n, M)-code over a q-ary alphabet Q, and let C_{in} be a binary (l, q)-code. Also, let ϕ denote a bijective mapping $\phi : Q \to C_{\text{in}}$ Then, the concatenated code C is the code obtained by considering each codeword $\mathbf{u} = (u_1, \ldots, u_n) \in C_{\text{out}}$ and mapping every symbol $u_i \in Q$ to a codeword $\phi(u_i) \in C_{\text{in}}$,

$$C = \{(\phi(u_1), \ldots, \phi(u_n)) : \mathbf{u} \in C_{\text{out}}\}.$$

The code C_{out} is called the *outer code* and C_{in} the *inner code*. The resulting code C is a binary $(n\,l, M)$-code with rate $R(C) = R(C_{\text{in}})R(C_{\text{out}})$.

Remark 2.14. Given n vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{F}_2^l$, the notation $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ above is a shorthand for the vector $(v_{11}, \ldots, v_{1l}, \ \ldots \ , v_{n1}, \ldots v_{nl}) \in \mathbb{F}_2^{nl}$.

We will usually deal with concatenation of codes where the outer code used in the construction is a linear code. Often, we will use Reed-Solomon or algebraic-geometric codes, which allows us to benefit from list decoding.

According to the discussion leading to Definition 2.12, a deterministic code is not sufficient, and some randomness over a family of codes is required. Code concatenation is a technique that has been used to obtain c-secure with ε-error families of codes with error probability decreasing exponentially with the code length, and equipped with efficient identification algorithms. See for example [13, 14, 15].

2.4 The Chernoff and Hoeffding Inequalities

We will have several occasions to use the following well-known results. Let $X_1, \ldots, X_n$ be n independent indicator r.v.'s, i.e., taking on values in $\{0, 1\}$. Also, let $X = \sum_1^n X_i$ and $p = E[n^{-1}X]$. That is, X counts the number of successes in n trials with average probability of success p. Then, the probabilities of the tails can be bounded as

$$\Pr\{n^{-1}X \geq p + \delta\} \overset{\text{(a)}}{\leq} 2^{-nD(p+\delta\|p)} \overset{\text{(b)}}{\leq} e^{-2n\delta^2}, \qquad \text{for } \delta > 0, \tag{2.10}$$

and

$$\Pr\{n^{-1}X \leq p - \delta\} \overset{\text{(a)}}{\leq} 2^{-nD(p-\delta\|p)} \overset{\text{(b)}}{\leq} e^{-2n\delta^2}, \qquad \text{for } 0 < \delta < p, \tag{2.11}$$

where $D(x\|y)$ denotes the Kullback-Leibler divergence between two Bernoulli distributed r.v.'s of parameters x and y, respectively,

$$D(x\|y) \overset{\text{def}}{=} x \log_2(x/y) + (1-x) \log_2((1-x)/(1-y)).$$

Inequalities (a) in (2.10) and (2.11) are known as the Chernoff bounds, and inequalities (b) are a special case of the Hoeffding bounds [35]. Observe that $D(x\|y) \geq 0$ and $D(x\|y) = 0$ if and only if $x = y$.

Remark 2.15. Obviously, the bound (2.10) holds for $p' \leq p$, and both (2.10) and (2.11) hold when X is a binomial r.v. of parameters n and p.

Chapter 3

Applications of Soft-Decision Decoding to Identify Traitors

The result of a collusion attack can be viewed as a "transmission through a noisy channel." In this case, the pirated word is a corrupted version of the traitor codewords. Intuitively, to identify some traitor, one has to "correct" a large number of errors. In [36,37] Silverberg, Staddon and Walker apply techniques that correct errors beyond the error-correction bound of the code to the identification process in IPP and TA codes.

The idea of correcting errors beyond the correcting capabilities of the code can be summarized as follows. In a code with minimum distance d, if in the transmission of a codeword the number of errors e is greater than $\left\lfloor \frac{d-1}{2} \right\rfloor$, then there can be more than one codeword within distance e from the received word. The decoder may either decode it incorrectly or fail to decode it. This leads to the concept of *list decoding* [38,39], where the decoder outputs a list of all codewords within distance $e > \left\lfloor \frac{d-1}{2} \right\rfloor$ of the received word, thus offering a potential way to recover from errors beyond the error-correction bound of the code. Although the concept of list decoding was proposed in the 1950's, for the case of Reed-Solomon codes no polynomial-time list-decoding algorithms were obtained until the breakthrough work presented by Sudan in 1997 [40]. For codes of rate greater than 1/3, the output list in the original work

of Sudan has size 1. However, Guruswami and Sudan overcome the rate limitation in another milestone paper [41, 42].

In *soft-decision decoding*, the input to the decoder is a reliability matrix that indicates, for each position, the probability that a given symbol from the alphabet was sent. Using this side information, the soft-decision decoder estimates the sent codeword. Building from the results of Guruswami and Sudan, in [43, 44] Kötter and Vardy present a polynomial-time soft-decision decoding algorithm for Reed-Solomon codes. This list-decoding algorithm is algebraic in nature and significantly outperforms both the Guruswami-Sudan decoding and the generalized minimum-distance decoding of Reed-Solomon codes.

In this chapter we discuss the application of the Kötter-Vardy soft-decision decoding algorithm for the identification process in TA codes, IPP codes and binary concatenated fingerprinting codes. The TA Tracing Algorithm consists in searching for a list of at most c codewords that contains parents of a given descendant. On the other hand, the IPP Tracing Algorithm consists in finding all coalitions of size at most c that can generate a given descendant. Finally, we deal with c-secure with ε-error families of binary codes. In this case, the construction that we discuss is based on code concatenation and the identification algorithm consists in searching for the codewords that can be identified with probability at least $1 - \varepsilon$ as parents of a given descendant. In all three cases, we take full advantage of the possibility of having as input a reliability matrix by making the Kötter-Vardy algorithm the core part of the identification process.

As said before, in [36, 37] the authors use the Guruswami-Sudan list-decoding algorithm in the TA and IPP Tracing Algorithms. However, in the case of the TA Tracing Algorithm their approach is only optimal when all parents (traitors) contribute equally to the construction of the pirated, and there is no guarantee to find more than one parent. In case of the IPP Tracing Algorithm, in order to find all coalitions that can generate a given pirated, they have to puncture the code.

In the TA Tracing Algorithm, we show how, by setting up the entries of the reliability matrix with appropriate values, traitors can be identified in polynomial time in the code length. For algebraic-geometric codes, a similar approach is made

in [45]. For the TA Tracing Algorithm, we also discuss an upper bound on the interpolation cost of the Kötter-Vardy algorithm.

In the case of the IPP Tracing Algorithm we present a straightforward algorithm that finds all coalitions capable of creating a given descendant. We discuss how, thanks to the Kötter-Vardy algorithm, the results in [36, 37] can be extensively improved.

To improve the rate of binary fingerprinting codes, many constructions [13, 14, 15] have used code concatenation [34], where the inner code is a binary code with some error probability ε. When the outer code is a Reed-Solomon code, we discuss generic constructions of such codes, equipped with an identification routine that uses the Kötter-Vardy algorithm. In this case, we will show that even a suboptimal setting of the entries of the Kötter-Vardy algorithm suffices for our purposes.

3.1 Reed-Solomon Codes and Soft-Decision Decoding

Reed-Solomon codes are a well-known class of linear codes [46, 47] that are used in a broad range of applications, ranging from CD encoding to satellite communications. They can be defined as follows.

Definition 3.1. Let γ be a primitive element of $\mathbb{F}_q$. The $[n, k]$-*Reed-Solomon code* over $\mathbb{F}_q$, of length $n = q - 1$ and dimension k, is defined as the following vector subspace of $\mathbb{F}_q^n$

$$\{(f(\gamma^1), \ldots, f(\gamma^n)) : f(x) \in \mathbb{F}_q[x],\ \deg f(x) < k\}.$$

Reed-Solomon codes are maximum distance separable (MDS) codes, since they meet the Singleton bound [48], and hence the $[n, k]$-Reed-Solomon code has minimum distance $d = n - k + 1$.

In this chapter, we will have occasion to develop some techniques that will work not only for the narrow-sense envelope model, but also for a model closely related

to the expanded narrow-sense envelope model from Definition 2.2, namely when no more than a threshold of erasures are allowed. In connection with this model, in [49] c-TA codes from Definition 2.9 are extended for the case of erasure tolerance.

Definition 3.2 ([49])**.** A code C is a *c-TA code tolerating s erasures* if for all subsets $U \subseteq C$ of size at most c, if $\mathbf{z} \in \text{desc}^*(U)$ with no more than s erasures, then there exists a $\mathbf{u} \in U$ such that $d(\mathbf{z}, \mathbf{u}) < d(\mathbf{z}, \mathbf{w})$ for all $\mathbf{w} \in C \setminus U$.

Also, the following result is a natural extension of (2.6) under the definition given above.

Theorem 3.3 ([49])**.** Let C be an $[n, k]$-code. If $d(C) > n(1 - 1/c^2) + s/c^2$, then C is a c-TA code tolerating s erasures.

From (2.7), a c-TA code is a c-IPP code. In general, the converse is false. However, it is conjectured that the converse is true for Reed-Solomon codes [36, 37]. We will elaborate more on this topic in Chapter 6.

Recall that, for a code with minimum distance d, if the number of errors e is greater than $\left\lfloor \frac{d-1}{2} \right\rfloor$, then there can be more than one codeword within distance e from the received word. In this situation, a list-decoding algorithm [38, 39, 41, 42, 43, 44] outputs a list of all codewords within distance e of the received word. In soft-decision decoding, the decoding process takes advantage of "side information," generated by the receiver and instead of using the received word symbols, the decoder uses probabilistic reliability information about these received symbols.

Before giving an overview of the Kötter-Vardy soft-decision decoding algorithm, let us introduce some concepts that will be useful below. For a detailed description see [43, 44].

A *discrete memoryless channel* can be defined as two finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, called the *input alphabet* and *output alphabet*, respectively, and $|\mathcal{X}|$ conditional pmf functions

$$f(y|x) \qquad \text{for all } x \in \mathcal{X},$$

where $y \in \mathcal{Y}$. We suppose that these pmf's are known to the decoder.

If we see the input and the output of a discrete memoryless channel as r.v.'s X and Y, respectively, and we suppose that X is uniformly distributed over $\mathcal{X}$, then the decoder can compute the probability that $\alpha_i \in \mathcal{X}$ was the transmitted symbol given that $\beta_j \in \mathcal{Y}$ was observed as

$$\Pr\{X = \alpha_i | Y = \beta_j\} = \frac{f(\beta_j|\alpha_i)}{\sum_{x \in \mathcal{X}} f(\beta_j|x)}. \tag{3.1}$$

For the case of Reed-Solomon codes, where the input alphabet is $\mathcal{X} = \mathbb{F}_q$, we take $\alpha_1, \alpha_2, \ldots, \alpha_q$ as the ordering of the elements of $\mathbb{F}_q$. If vector $\beta = (\beta_1, \ldots, \beta_n)$ is received, then using (3.1) the following values can be computed:

$$r_{ij} \stackrel{\text{def}}{=} \Pr\{X = \alpha_i | Y = \beta_j\}. \tag{3.2}$$

These values are the entries of a stochastic q-by-n matrix $\mathcal{R}$, called the *reliability matrix*, which is the input of the Kötter-Vardy algorithm. This matrix is then transformed into a q-by-n *multiplicity matrix* $\mathcal{M}$, used in the subsequent steps of the algorithm.

We are interested in knowing what codewords the Kötter-Vardy algorithm (Algorithm 3.1) will return. With this aim, given two q-by-n matrices $A = (a_{ij})$ and $B = (b_{ij})$ over the same field, the following product is defined:

$$\langle A, B \rangle \stackrel{\text{def}}{=} \operatorname{trace}(AB^T) = \sum_{i=1}^{q} \sum_{j=1}^{n} a_{ij} b_{ij}. \tag{3.3}$$

Moreover, a word $\mathbf{u} = (u_1, \ldots, u_n)$ over $\mathbb{F}_q$ can be represented by an q-by-n matrix $[\mathbf{u}] = ([\mathbf{u}]_{ij})$. The entries $[\mathbf{u}]_{ij}$ are defined as follows:

$$[\mathbf{u}]_{ij} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } u_j = \alpha_i, \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 3.1 briefly outlines the Kötter-Vardy algorithm. For a detailed description, see [43, 44]. The algorithm makes use of the following notion.

Algorithm 3.1 The Kötter-Vardy soft-decision decoding algorithm

Initial ordering of the elements of $\mathbb{F}_q$: $\alpha_1, \alpha_2, \ldots, \alpha_q$.

Input: An $[n, k]$-Reed-Solomon code C over $\mathbb{F}_q$ and a q-by-n reliability matrix $\mathcal{R}$.

Output: A subset of codewords of C.

KV$[C, \mathcal{R}]$:

1) Using a multiplicity-assignment algorithm, transform the reliability matrix $\mathcal{R}$, into nonnegative q-by-n matrix of integers $\mathcal{M}$, called multiplicity matrix, so that $\mathcal{M}$ maximizes the expectation of $\langle \mathcal{M}, [\mathbf{u}] \rangle$, where $\mathbf{u}$ is the transmitted codeword.

2) *Interpolation step:* From the multiplicity matrix $\mathcal{M} = (m_{ij})$, compute a bivariate polynomial $Q(x, y)$ with minimum $(1, k-1)$-weighted degree such that for every nonzero m_{ij} it has a zero at the point (γ_j, α_i) of multiplicity at least m_{ij}. Here, γ_j is the evaluation point of the Reed-Solomon code at the jth position.

3) *Factorization step:* Find all the univariate polynomials $f(x)$ such that $(y - f(x))$ divide $Q(x, y)$. The output of the algorithm are the codewords generated by every such $f(x)$.

Definition 3.4. Let $Q(x,y) = \sum_{ij} q_{ij} x^i y^j$ be a bivariate polynomial in $\mathbb{F}_q[x,y]$. The (w_x, w_y)-*weighted degree* of $Q(x,y)$ is defined as

$$\max\{w_x i + w_y j : q_{ij} \neq 0\}.$$

It is worth noting here that in the interpolation step each interpolation point m_{ij} imposes a set of $m_{ij}(m_{ij}+1)/2$ linear constraints on the construction of $Q(x,y)$. Hence, for a given multiplicity matrix $\mathcal{M} = (m_{ij})$, the total number of linear constraints, denoted $\text{cost}(\mathcal{M})$, imposed on the interpolation polynomial $Q(x,y)$ is

$$\text{cost}(\mathcal{M}) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{q} \sum_{j=1}^{n} m_{ij}(m_{ij}+1). \tag{3.4}$$

This value is referred in the literature as the *interpolation cost*, since it has a direct impact in both the outcome and the runtime of the interpolation process.

In [44] Kötter and Vardy state the following theorem.

Theorem 3.5 ([44])**.** Let C be an $[n,k]$-Reed-Solomon code, and let $\alpha_1, \alpha_2, \ldots, \alpha_q$ be an ordering of the elements of $\mathbb{F}_q$. If the codeword $\mathbf{u} \in C$ is transmitted, and the word $\mathbf{v} = (v_1, \ldots, v_n)$ is received and the reliability matrix $\mathcal{R} = (r_{ij})$ is constructed according to (3.2),

$$r_{ij} = \Pr\{X = \alpha_i | Y = v_j\},$$

then the Kötter-Vardy soft-decision decoding algorithm outputs a list that contains the transmitted codeword $\mathbf{u}$ if

$$\frac{\langle \mathcal{R}, [\mathbf{u}] \rangle}{\sqrt{\langle \mathcal{R}, \mathcal{R} \rangle}} \geq \sqrt{k-1} + o(1), \tag{3.5}$$

where $o(1)$ is a function that tends to zero when the number of interpolation points counted with multiplicities (and hence, the interpolation cost) is taken to infinity.

3.1.1 Performance of the Kötter-Vardy Algorithm for the q-ary Symmetric Channel

The performance of the Kötter-Vardy algorithm can be improved for certain channels including the *q-ary symmetric erasure channel*.

A q-ary symmetric erasure channel with error probability ξ, erasure probability δ, input alphabet $\mathcal{X} = \mathbb{F}_q$ and output alphabet $\mathcal{Y} = \mathbb{F}_q \cup \{*\}$, can be characterized as an $|\mathcal{X}|$-by-$|\mathcal{Y}|$ *transition probability matrix* $W_{Y|X}$. If the rows are indexed by $\mathcal{X}$, and the columns by $\mathcal{Y}$, then the transition probability matrix $W_{Y|X}$ has the following expression:

$$W_{Y|X}(x,y) = \begin{cases} \delta & \text{if } y = *, \\ (1-\delta)(1-\xi) & \text{if } y = x, \\ (1-\delta)\dfrac{\xi}{q-1} & \text{otherwise.} \end{cases}$$

To construct the reliability matrix $\mathcal{R}$, suppose that codeword $\mathbf{u}$ is transmitted and word $\mathbf{v}$ is received. For this particular channel we have

$$\mathcal{R} = (1-\xi)[\mathbf{v}] + \frac{\xi}{q-1}(\mathbf{1} - [\mathbf{v}]),$$

where, in this case, $[\mathbf{v}]$ is the q-by-n matrix defined as

$$[\mathbf{v}]_{ij} \stackrel{\text{def}}{=} \begin{cases} 1/q & \text{if } v_j = *, \\ 1 & \text{if } v_j = \alpha_i, \\ 0 & \text{otherwise,} \end{cases}$$

and $\mathbf{1}$ denotes the q-by-n all-one matrix.

If we suppose that $n-m$ erasures and $m-l$ errors occurred during the transmission (i.e., l correct symbols received), then, using the matrix product defined in (3.3) and Theorem 3.5, the Kötter-Vardy algorithm will output codeword $\mathbf{u}$ if

$$\frac{l(1-\xi) + (m-l)\frac{\xi}{q-1} + \frac{n-m}{q}}{\sqrt{m(1-\xi)^2 + m\frac{\xi^2}{q-1} + \frac{n-m}{q}}} > \sqrt{k-1}. \tag{3.6}$$

Below we will need to maximize the number of errors that the Kötter-Vardy algorithm can correct. To make matters worse, in the settings in which we will use the Kötter-Vardy algorithm the channel parameter ξ will be unknown. This is due to the fact that ξ will depend on the strategy of a coalition of traitors performing a collusion attack. Therefore, given the code parameters, we are free to choose the value of ξ and it is clearly convenient to choose the one that maximizes the left-hand side of (3.6). According to [50], intuitively, this corresponds to maximizing the left-hand side of (3.6) with respect to the worst channel that the Kötter-Vardy algorithm can still handle for a given code rate. This value is

$$\xi = \frac{m-l}{m}. \tag{3.7}$$

For this value of ξ, equation (3.6) remains valid and it reduces to

$$\frac{l^2}{m} + \frac{(m-l)^2}{m(q-1)} + \frac{n-m}{q} > k-1. \tag{3.8}$$

This means that if upon receiving a word $\mathbf{v}$, with $n-m$ symbols erased, then for every value of l that satisfies (3.8) the Kötter-Vardy algorithm will output the transmitted codeword $\mathbf{u}$. Therefore the algorithm can handle $n-m$ erasures and $m-l$ errors.

3.2 The TA Tracing Algorithm

In this section we focus on the identification process of TA Reed-Solomon codes using the Kötter-Vardy algorithm.

For a c-TA Reed-Solomon code tolerating s erasures, the goal of the TA Tracing Algorithm is to output a list of size at most c that contains as many parents of a given descendant as possible. One cannot expect to find all parents, since some of them may contribute with too few positions and cannot be identified. This happens, for example, when a parent contributes with no more than $k-1$ positions.

We begin with the following proposition.

Proposition 3.6. Let C be a c-TA (n, M)-code with minimum distance $d = d(C)$ tolerating s erasures, and let $\mathbf{z} \in \mathrm{desc}_c^*(C)$ be a descendant of some coalition of size at most c. If a codeword $\mathbf{u} \in C$ agrees in at least $c(n-d)+1$ unerased positions with $\mathbf{z}$, then $\mathbf{u}$ belongs to all coalitions of size at most c that are able to generate $\mathbf{z}$:

$$\mathbf{u} \in \bigcap_{\substack{U \subseteq C, |U| \leq c, \\ \text{s.t. } \mathbf{z} \in \mathrm{desc}^*(U)}} U.$$

Proof. If the code C has minimum distance d, then two codewords can agree in at most $n-d$ positions. Therefore a coalition of size c is able to create a descendant $\mathbf{z}$ that agrees in at most $c(n-d)$ positions with any other codeword outside the coalition. Hence, if there exists a codeword $\mathbf{u}$ that agrees with $\mathbf{z}$ in at least $c(n-d)+1$ positions, then this codeword must be unique. Therefore $\mathbf{u}$ belongs to all coalitions of size at most c that are able to create $\mathbf{z}$. □

Corollary 3.7. Let C be a c-TA (n, M)-code with minimum distance $d = d(C)$ tolerating s erasures, and let $\mathbf{z} \in \mathrm{desc}_c^*(C)$ be a descendant of some coalition of size at most c. Furthermore, assume that $\mathbf{z}$ has at most s positions erased. Let $\mathbf{u}^1, \ldots, \mathbf{u}^j$ be $j < c$ already identified parents that lie in the intersection of all coalitions of size at most c that are able to create $\mathbf{z}$:

$$\mathbf{u}^i \in \bigcap_{\substack{U \subseteq C, |U| \leq c, \\ \text{s.t. } \mathbf{z} \in \mathrm{desc}^*(U)}} U, \qquad \text{for all } 1 \leq i \leq j.$$

If these j parents jointly match less than $n - s - (c-j)(n-d)$ positions of $\mathbf{z}$, then any codeword $\mathbf{u} \neq \mathbf{u}^1, \ldots, \mathbf{u}^j$ that agrees with $\mathbf{z}$ in at least $(c-j)(n-d)+1$ of the yet unmatched positions also lies in the intersection.

The previous corollary motivates the following definition.

Definition 3.8. Let C be a c-TA code tolerating s erasures and let $\mathbf{z} \in \mathrm{desc}_c^*(C)$ be a descendant of some coalition of size at most c. We call the set of codewords satisfying the conditions of Proposition 3.6 and Corollary 3.7 the *set of TA-parents* of $\mathbf{z}$, denoted $\mathcal{P}_{\mathrm{TA}}(\mathbf{z})$.

Algorithm 3.2 TA Tracing Algorithm

Initial ordering of the elements of $\mathbb{F}_q$: $\alpha_1, \alpha_2, \ldots, \alpha_q$.

Input:

- c: maximum size of the coalition,
- s: maximum number of erased positions,
- C: an $[n, k]$-Reed-Solomon code with minimum distance $d > n(1-1/c^2)+s/c^2$,
- $\mathbf{z}$: a descendant in $\mathrm{desc}_c^*(C)$ with $\leq s$ positions erased.

Output: A list of all TA-parents of $\mathbf{z}$, $\mathcal{P}_{\mathrm{TA}}(\mathbf{z})$.

TA$[c, s, C, \mathbf{z}]$:

1) Initially set

$$i := 1, \qquad c_i := c, \qquad S_i := \{t : z_t = *\}, \qquad L := \emptyset.$$

2) Compute the q-by-n reliability matrix

$$\mathcal{R} := (1-\xi)[\mathbf{z}] + \frac{\xi}{q-1}(\mathbf{1} - [\mathbf{z}]),$$

where, for $1 \leq a \leq q$ and $1 \leq b \leq n$,

$$[\mathbf{z}]_{a,b} := \begin{cases} 1/q & \text{if } b \in S_i, \\ 1 & \text{if } b \notin S_i \text{ and } z_b = \alpha_a, \\ 0 & \text{otherwise,} \end{cases}$$

using the error parameter $\xi := \frac{n-|S_i|-l}{n-|S_i|}$, with

$$l := \max\left\{c_i(n-d)+1, \left\lceil \frac{n-|S_i|}{c_i} \right\rceil \right\}.$$

3) Plug the matrix $\mathcal{R}$ into the Kötter-Vardy algorithm and, from the output list, take the set $\Lambda = \{\mathbf{u}^1, \ldots, \mathbf{u}^j\}$ of all codewords that agree with $\mathbf{z}$ in, at least, $c_i(n-d)+1$ positions not in S_i. Set $L := L \cup \Lambda$.

4) Set

$$\begin{aligned} i &:= i+1, \\ c_i &:= c_{i-1} - j, \\ S_i &:= \{t : z_t = *\} \cup \{t : z_t = u_t \text{ for some } \mathbf{u} \in L\}. \end{aligned}$$

5) If $j = 0$ or if $c_i = 0$ or if $|S_i| \geq n - c_i(n-d)$, output L and quit. Else go to Step 2).

Based on the Kötter-Vardy algorithm, the key idea of the TA Tracing Algorithm (Algorithm 3.2) is described in Corollary 3.7. Given a c-TA Reed-Solomon code tolerating s erasures, and given a descendant $\mathbf{z} \in \mathrm{desc}_c^*(C)$, there is no side information available. Hence, in the first iteration, the Kötter-Vardy algorithm is executed, constructing the reliability matrix as if the channel were a q-ary erasure channel. The error parameter is computed according to (3.6) and (3.7). When some TA-parents are identified, their matching positions with the descendant $\mathbf{z}$ are treated as erased positions. Again, the reliability matrix and the error parameter are computed, now considering the minimum number of positions where a TA-parent and the descendant must agree to be declared a positive TA parent, and the Kötter-Vardy algorithm is executed. The process continues until it becomes clear that there are no more TA-parents.

3.2.1 Correctness of the Algorithm

As mentioned above, we construct the reliability matrix as if the channel were a q-ary symmetric erasure channel. This type of channel is memoryless by definition. Unfortunately, in their attack, the traitors are free to use any strategy of their choice. In particular, they can compute an output symbol based on the entire set of detected symbols. For instance, equal contribution of symbols from each traitor to the descendant can be seen as a strategy with memory. Nevertheless, even if memory is used by the traitors, once a descendant has been created, the positions in which this descendant and a TA-parent disagree can be treated as "errors in the transmission." In the rest of this section, we show that ignoring the ability of the traitors to use memory is completely safe for our purposes.

Initially, the "errors in the transmission" are the number of unerased positions where a TA-parent and the descendant differ. Moreover, the set of "erased positions" S_1 contains the s positions that have been erased from the descendant. Since the minimum distance of the $[n, k]$-Reed-Solomon code is $d > n(1 - 1/c^2) + s/c^2$, and $k - 1 = n - d$, one can check from (3.6) that L is not empty after the first iteration. This is because one TA-parent matches the descendant in at least m/c positions. In

iteration $i > 1$ the set of erased positions is virtually augmented with the positions where the descendant coincides with some previously identified TA-parent. Note that in this iteration no TA-parent will be identified if $|S_i| \geq n - c_i(n-d)$, and also note that, at least, one TA-parent will be identified if $|S_i| < n - c_i^2(n-d)$. This leaves open an uncertainty interval,

$$n - c_i^2(n-d) \leq |S_i| < n - c_i(n-d),$$

where the algorithm will output a codeword whenever its contribution satisfies the condition of Corollary 3.7. Note also that, within the uncertainty interval, the TA Tracing Algorithm executes the Kötter-Vardy algorithm with $l \geq c_i(n-d)+1$ and hence, the decoding radius satisfies the condition of Corollary 3.7.

Lemma 3.9. The TA Tracing Algorithm identifies all TA-parents of a given descendant.

Proof. We have to show that no TA-parent remains unidentified when the algorithm reaches a terminating condition. From Step 5) it is clear that the algorithm terminates in one of the following three cases: when $j = 0$, or $c_i = 0$ or $|S_i| \geq n - c_i(n-d)$.

If $c_i = 0$, then we have $|L| = c$. This means that there are no unidentified TA-parents.

If $|S_i| \geq n - c_i(n-d)$, then by Corollary 3.7 there cannot be any other TA-parent.

Now, it is only left to show that if in iteration i there are still unidentified TA-parents, then at least one of them will appear in the output list of the Kötter-Vardy algorithm. In other words, we will have that $j > 0$. Following the notation of Section 3.1.1, we denote by $m_i = n - |S_i|$ the number of "unerased positions" in iteration i, and by l the number of "correct positions," i.e., the number of unerased positions where a TA-parent and the descendant agree. In iteration i there can be at most $c_i = c - i + 1$ unidentified parents. We first suppose that $m_i \leq c_i^2(n-d)$. A TA-parent is a codeword that coincides with the descendant in $l \geq c_i(n-d)+1$ unerased positions. For every TA-parent we have

$$\frac{l^2}{m_i} \geq \frac{(c_i(n-d)+1)^2}{c_i^2(n-d)} > k - 1 + \frac{1}{c_i}.$$

It follows that (3.8) is satisfied and, as a consequence, all TA-parents are returned by the Kötter-Vardy algorithm and can be identified. Now, suppose that the number of unerased positions is $m_i > c_i^2(n-d)$. In this case, there exists a TA-parent such that $l > m_i/c_i$. For this particular TA-parent, we have $l^2/m_i > m_i/c_i^2 > k-1$. Again, it follows that (3.8) is satisfied and therefore this TA-parent is identified. □

3.2.2 Bounding the Interpolation Cost

In the TA Tracing Algorithm above, we have focused on the setup of the reliability matrix without taking into account the insights of the Kötter-Vardy algorithm. We have shown that all TA-parents can be identified, but at the expense of a very large and undetermined cost. In this section, we propose how to bound the interpolation cost for a practical execution of the Kötter-Vardy algorithm, ensuring that the set of TA-parents are still contained in the output list.

The condition for successful identification in the TA Tracing Algorithm is based on Theorem 3.5. Equation (3.5) implies an asymptotic performance of the Kötter-Vardy algorithm, so we have not been paying any attention to the cost of the algorithm. To introduce the cost into the discussion, we recall from Section 3.1 that the interpolation cost is the total number of linear restrictions imposed on the interpolation polynomial, computed according to (3.4).

Suppose that codeword $\mathbf{u}$ is a TA-parent. Then, according to [44], the Kötter-Vardy algorithm will return a list of codewords that contains $\mathbf{u}$ if the computed multiplicity matrix $\mathcal{M}$ satisfies

$$\langle \mathcal{M}, [\mathbf{u}] \rangle \geq \sqrt{2(k-1)\,\mathrm{cost}(\mathcal{M})}, \tag{3.9}$$

which for asymptotically large costs is reduced to (3.5). The multiplicity matrix $\mathcal{M}$ is obtained from $\mathcal{R}$ using a multiplicity-assignment scheme that, for every real value λ, allows to express $\mathcal{M} = \lfloor \lambda \mathcal{R} \rfloor$. We take advantage of this expression to obtain a bound for which (3.9) is satisfied.

Assuming that we construct the multiplicity matrix as $\mathcal{M} = \lfloor \lambda \mathcal{R} \rfloor$, being $\mathbf{u}$ a TA-parent in iteration i of the TA Tracing Algorithm, the left-hand side of (3.9) is

$$\left\lfloor \frac{\lambda}{q} \right\rfloor |S_i| + \lfloor \lambda(1-\xi) \rfloor l + \left\lfloor \frac{\lambda \xi}{q-1} \right\rfloor (n - |S_i| - l), \tag{3.10}$$

and the interpolation cost is

$$\text{cost}(\mathcal{M}) = q|S_i| \tbinom{\lfloor \lambda/q \rfloor + 1}{2} + (n - |S_i|) \left[\tbinom{\lfloor \lambda(1-\xi) \rfloor + 1}{2} + (q-1) \tbinom{\lfloor \lambda \xi/(q-1) \rfloor + 1}{2} \right], \tag{3.11}$$

which, for reasonable values of the code parameters, is always upper bounded as

$$\text{cost}(\mathcal{M}) \leq 3(n - |S_i|) \binom{\lfloor \lambda(1-\xi) \rfloor + 1}{2}. \tag{3.12}$$

Since the cost is an increasing function of λ, we are interested in finding the minimum value of λ such that (3.9) is satisfied. This is equivalent to define the function

$$f(\lambda) \stackrel{\text{def}}{=} \langle \lfloor \lambda \mathcal{R} \rfloor, [\mathbf{u}] \rangle^2 - 2(k-1)\,\text{cost}(\mathcal{M}) \tag{3.13}$$

and find a bound λ' such that $f(\lambda) \geq 0$ for any $\lambda \geq \lambda'$. One can always determine such λ' by direct search. Note that we only have to test the values of λ that change the values of the floor functions of $f(\lambda)$. Such values are of the form

$$\sum_{\substack{k \in \mathbb{Z} \\ b_1, b_2, b_3 \in \{0,1\}}} k q^{b_1} \left(\frac{1}{1-\xi} \right)^{b_2} \left(\frac{q-1}{\xi} \right)^{b_3}.$$

A more straightforward approach to determine a bound for λ' is presented in the following lemma.

Lemma 3.10. Let $g(\lambda)$ be the degree-2 polynomial constructed as

$$g(\lambda) = A^2 - 2(k-1)B,$$

where A is the expression obtained by substituting the floor functions $\lfloor x \rfloor$ by $x - 1$ in (3.10) and B is the expression obtained by substituting the floor functions $\lfloor x \rfloor$ by x in (3.11), and let $\{\lambda_1, \lambda_2\}$ be the set of roots of $g(\lambda)$. Then $\lambda' \leq \max\{\lambda_1, \lambda_2\}$.

Proof. Note that $g(\lambda)$ is a degree-2 polynomial lower bound of the function $f(\lambda)$ defined in (3.13), with positive leading coefficient. Therefore, its largest root must occur beyond the point where $f(\lambda)$ becomes positive. □

Hence, the cost of the interpolation process to find a TA-parent in the TA Tracing Algorithm is upper bounded by (3.12) substituting λ by the maximum root of the polynomial defined in Lemma 3.10. Note that the TA Tracing Algorithm loops at most c times. Therefore, the overall interpolation cost of the algorithm can be upper bounded by

$$\text{cost}(\mathcal{M}) \leq \frac{3}{2} c(n - |S|)\lambda(\lambda + 1), \tag{3.14}$$

taking the value of λ from Lemma 3.10. In other words, all TA-parents will be identified with a total interpolation cost given by (3.14), and one cannot expect to identify more TA-parents even allowing the instances of the Kötter-Vardy algorithm run with a global interpolation cost higher than that.

3.3 The IPP Tracing Algorithm

In this section, we focus on the use of the Kötter-Vardy algorithm as the underlying routine for the IPP identification process in Reed-Solomon codes.

From Definition 2.8, in a c-IPP code all coalitions of size at most c that are able to generate a given descendant have a non-empty intersection. Clearly, the codewords that lie in the intersection are the only ones that can be accused with certainty as traitors.

Definition 3.11. Let C be a c-IPP code and let $\mathbf{z} \in \text{desc}_c(C)$ be a descendant of some coalition of size at most c. We define the *set of IPP-parents of* $\mathbf{z}$, denoted by $\mathcal{P}_{\text{IPP}}(\mathbf{z})$, as the set of codewords of C that belong to all coalitions of size at most c

that are able to generate $\mathbf{z}$:

$$\mathcal{P}_{\mathrm{IPP}}(\mathbf{z}) \stackrel{\mathrm{def}}{=} \bigcap_{\substack{U \subseteq C, |U| \leq c, \\ \text{s.t. } \mathbf{z} \in \mathrm{desc}(U)}} U.$$

The proof of next lemma is immediate from the definitions.

Lemma 3.12. Let C be a c-TA code and let $\mathbf{z} \in \mathrm{desc}_c(C)$ be a descendant of some coalition of size at most c. Then $\mathcal{P}_{\mathrm{TA}}(\mathbf{z}) \subseteq \mathcal{P}_{\mathrm{IPP}}(\mathbf{z})$.

Therefore, determining the set of c-IPP parents of a c-IPP code consists in searching for coalitions of size at most c. If the code has M codewords, this task has a runtime complexity of $O\binom{M}{c}$. Below we discuss an identification algorithm for c-IPP Reed-Solomon codes based on list decoding.

As mentioned in Section 3.1 the characterization of c-IPP Reed-Solomon codes is not clear. Fortunately, using the proven fact that any c-TA code is a c-IPP code, a Reed-Solomon code that satisfies the distance condition (2.6) will suffice for our purposes. Note also that, for any c-IPP code, there is more than one coalition that can generate a given descendant $\mathbf{z}$ only if $|\mathcal{P}_{\mathrm{IPP}}(\mathbf{z})| < c$.

Before discussing the IPP Tracing Algorithm (Algorithm 3.3) at length, we first give some intuition. The algorithm that we present is recursive in nature. It receives as its input a list of codewords L that (partially) "cover" $\mathbf{z}$. Then, for this received input list, the algorithm looks for an appropriate set of candidate codewords that cover positions not already covered by the codewords in L. For each one of these candidates $\mathbf{u}$, the algorithm executes recursively, now using $L \cup \{\mathbf{u}\}$ as its input list. Clearly, this process eventually returns all coalitions that can generate $\mathbf{z}$ if a list that contains a subset of the IPP-parents is given in the initial call of the algorithm. This can be accomplished according to Lemma 3.12 by using, for example, the TA Tracing Algorithm discussed in Section 3.2.

Also, as opposed to the case of the TA Tracing Algorithm, list decoding cannot offer a total solution to the IPP identification problem. This is immediate to see by the following simple example. Take a 2-IPP $[n, k]$-Reed-Solomon code. If a descendant $\mathbf{z}$ contains $n-1$ symbols from a given parent, say $\mathbf{u}$, then there are q^{k-1} possibilities for

Algorithm 3.3 IPP Tracing Algorithm

Initial ordering of the elements of $\mathbb{F}_q$: $\alpha_1, \alpha_2, \ldots, \alpha_q$.
A global variable $\mathcal{L}$ is needed. Initially set $\mathcal{L} = \emptyset$.
The initial call needs to be with $L := \mathrm{TA}[c, 0, C, \mathbf{z}]$.

Input:

- c: maximum size of the coalition,
- C: an $[n, k]$-Reed-Solomon code with minimum distance $d > n(1 - 1/c^2)$,
- $\mathbf{z}$: a descendant in $\mathrm{desc}_c(C)$,
- L: a (partial) list of parents of $\mathbf{z}$.

Output: The set $\mathcal{L}$ of all coalitions $L \subseteq C$ with $|L| \leq c$ such that $\mathbf{z} \in \mathrm{desc}(L)$.

IPP$[c, C, \mathbf{z}, L]$:

1) $S := \{t : z_t = v_t \text{ for some } \mathbf{u} \in L\}$.
   - If $|S| = n$, then set $\mathcal{L} := \mathcal{L} \cup \{L\}$ and quit.
   - If $|S| < n$ and $|L| = c$, then quit.
2) Compute the q-by-n reliability matrix

$$\mathcal{R} := (1 - \xi)[\mathbf{z}] + \frac{\xi}{q-1}(\mathbf{1} - [\mathbf{z}]),$$

   where, for $1 \leq a \leq q$ and $1 \leq b \leq n$,

$$[\mathbf{z}]_{a,b} := \begin{cases} 1/q & \text{if } b \in S, \\ 1 & \text{if } b \notin S \text{ and } z_b = \alpha_a, \\ 0 & \text{otherwise,} \end{cases}$$

   using the error parameter $\xi := 1 - \frac{l}{n-|S|}$, with $l := \left\lceil \frac{n-|S|}{c-|L|} \right\rceil$.
3) Plug the matrix $\mathcal{R}$ into the Kötter-Vardy algorithm and, from the output list, take the set $\Lambda = \{\mathbf{u}^1, \ldots, \mathbf{u}^s\}$ of all codewords that agree with $\mathbf{z}$ in at least l positions not in S.
4) If $\Lambda = \emptyset$,
   Reencoding step:
   - Set $j := \min\{l, k\}$.
   - For each subset $\{t_1, \ldots, t_j\}$ of j positions of $\mathbf{z}$ not in S,
     - If $j = k$,
       - $\mathbf{v} := \mathbf{reencode}[(z_{t_1}, t_1), \ldots, (z_{t_k}, t_k)]$,
       - $\Lambda := \Lambda \cup \{\mathbf{v}\}$.
     - Else,
       - Fix $k - j$ positions $t_{j+1}, \ldots, t_k$ in S.
       - For all $(x_1, \ldots, x_{k-j}) \in \mathbb{F}_q^{k-j}$,
         $\mathbf{v} := \mathbf{reencode}[(z_{t_1}, t_1), \ldots, (z_{t_j}, t_j), (x_1, t_{j+1}), \ldots, (x_{k-j}, t_k)]$,
         $\Lambda := \Lambda \cup \{\mathbf{v}\}$.
5) For each $\mathbf{u} \in \Lambda$, execute IPP$[c, C, \mathbf{z}, L \cup \{\mathbf{u}\}]$.

the remaining parent. Moreover, in this case the Kötter-Vardy algorithm should be able to correct $n-1$ erasures. From (3.8) it is clear that this is not possible. When faced with this situation, we use reencoding in the style of [51] in order to find the remaining codewords that can be part of a coalition.

3.3.1 Considerations about the Reencoding Step

In Step 3) of the IPP Tracing Algorithm, if the list returned by the Kötter-Vardy algorithm is empty or its elements do not cover any position not in S, then we must devise a different method to find the remaining elements to complete the coalitions. As discussed above, the method that we use is based on reencoding [51]. Due to the MDS property, for a Reed-Solomon code of dimension k, we can treat the symbols in any k index positions $t_1, \ldots, t_k$ as information symbols. This allows us to encapsulate the encoding steps into a routine **reencode**$[(z_1, t_1), \ldots, (z_k, t_k)]$, where $z_1, \ldots, z_k$ are variables that take values from the elements of $\mathbb{F}_q$. Therefore, **reencode**$[(z_1, t_1), \ldots, (z_k, t_k)]$ returns the unique codeword with symbols $z_1, \ldots, z_k$ in positions $t_1, \ldots, t_k$, respectively. In the case of Reed-Solomon codes this can be achieved easily by the evaluation of a Lagrange-interpolation polynomial.

Assume that c_i is the number of remaining traitors to complete a coalition. There are two different cases to be considered. The first case is when $l = \lceil (n-|S|)/c_i \rceil \geq k$. In this case, we can assume that at least one remaining codeword can be found by taking all of its information positions not in S. This is again due to the MDS property of Reed-Solomon codes. To do so, the algorithm runs over all the possible subsets of size k among the $n-|S|$ unerased positions and by applying the reencoding routine to each subset obtains the corresponding codeword. Note that the maximum number of reencodings in this case is upper bounded by $\binom{n-|S|}{k}$.

On the other hand, we can have that $l = \lceil (n-|S|)/c_i \rceil < k$. In this case, we cannot assume that any remaining coalition codeword agrees in k unerased positions with the descendant. Suppose that for l positions outside S say $t_1, \ldots, t_l$, the descendant $\mathbf{z}$ has symbols $z_{t_1}, \ldots, z_{t_l}$, respectively. Hence, we need to search for all codewords that agree with $\mathbf{z}$ in these l positions. To do so, we fix a set of $k-l$ positions in S that

we represent as $t_{l+1}, \ldots, t_k$. Let the variable $(x_1, \ldots, x_{k-l})$ take all possible values from $\mathbb{F}_q^{k-l}$. Then, for each value of $(x_1, \ldots, x_{k-l})$ the reencoding routine with input $(z_{t_1}, t_1), \ldots, (z_{t_l}, t_l), (x_1, t_{l+1}), \ldots, (x_{k-l}, t_k)$ will return one of the desired codewords. In this case, the maximum number of reencodings is upper bounded by $\binom{n-|S|}{l} q^{k-l}$.

Whenever one codeword is found, the remaining codewords to be added to a coalition, if any, are found by recursive executions of the algorithm. Below, we present a lemma that proves that, following this procedure, the output of the algorithm $\mathcal{L}$ will eventually contain all the lists of codewords of size at most c that are able to generate a given descendant.

3.3.2 Correctness of the Algorithm

Lemma 3.13. The IPP Tracing Algorithm identifies all coalitions of size at most c that can generate a given descendant.

Proof. Note that the algorithm is executed recursively. Let L be the starting set of codewords at a certain invocation of the algorithm. We first show that if there is a coalition $L' \subseteq C$ with $|L'| \leq c$ that can generate $\mathbf{z}$ and $L \subseteq L'$, then L' will eventually be included in the global set $\mathcal{L}$. If $L = L'$, it is obvious that L' will be included in $\mathcal{L}$ in Step 1). Otherwise, by the pigeonhole principle, there is a codeword $\mathbf{u} \in L' \setminus L$ such that it agrees with $\mathbf{z}$ in, at least, $l = (n - |S|)/(c - |L|)$ positions not in S. In Steps 2) – 4) the algorithm identifies such codeword, either using the Kötter-Vardy algorithm or in the reencoding step. Now, the algorithm is executed again using as input $L_1 = L \cup \{\mathbf{u}\}$. It is clear that $L \subset L_1 \subseteq L'$. Again, since L' can generate $\mathbf{z}$, then either $L_1 = L'$ or we can find a subset L_2 such that $L_1 \subset L_2 \subseteq L'$. Because $|L'| \leq c$, there is only a finite number of subsets, say $s \leq c$, such that $L \subset L_1 \subset L_2 \subset \cdots \subset L_s \subset L'$. Therefore, the algorithm will eventually be executed with L' as input and, hence, L' will be included in $\mathcal{L}$.

On the other hand, observe that the initial call of the algorithm is executed using the set of TA-parents, generated by the TA Tracing Algorithm. This set belongs to all coalitions that can generate $\mathbf{z}$.

It follows that when the recursive executions of the algorithm finish, $\mathcal{L}$ will contain all coalitions $L \subseteq C$ of size at most c that can generate $\mathbf{z}$. □

To determine the running complexity of the IPP Tracing Algorithm, let c' be the size of the list returned by the TA Tracing Algorithm.

We first consider the case that at each recursion the execution of the Kötter-Vardy algorithm is successful. In other words, we obtain the list Λ of all codewords that agree in at least l positions with the descendant. Then the total number of recursions is upper bounded by $|\Lambda|^{c-c'-1}$, with $|\Lambda| < c$. Therefore, the running time is $O(|\Lambda|^{c-c'-1}T_{\mathrm{KV}})$, where T_{KV} denotes the complexity of the Kötter-Vardy algorithm, which is polynomial in the code length. Hence, the algorithm offers a considerable improvement over both the brute force approach and the algorithm presented in [36, 37].

On the other hand, it might be the case that the Kötter-Vardy algorithm does not return any appropriate codeword. If we take as the worst-case situation when the Kötter-Vardy algorithm fails in each recursion, then of course there is not much room for improvement. In this case the number of executions of the IPP Tracing Algorithm will be upper bounded by $\binom{M}{c-c'}$, i.e., an execution time $O(M^{c-c'})$, as noted in [36,37]. This is, however, a clear improvement over the brute-force method, since $c' \geq 1$.

3.4 Concatenated Constructions

In this section, we deal with the case of c-secure with ε-error families of binary codes. As said in Section 2.2, in the binary case the identification process will always be subjected to a certain error probability. To construct practical (shorter) binary fingerprinting codes many authors [13, 14, 15, 33] have used the idea of code concatenation [34].

According to the discussion leading to Definition 2.12, a single binary code is not sufficient, but a family of codes $\mathcal{C} = \{C_t\}_{t \in T}$ is required. We now show how to obtain a family of binary concatenated fingerprinting codes with error probability

decreasing exponentially with the code length, by modifying the codes obtained in Construction 2.13.

Construction 3.14. Let C_{out} be an (n, M)-code over a q-ary alphabet Q. Rather than a single inner code, let $\mathcal{C}_{\text{in}} = \{C_s^{\text{in}}\}_{s \in S}$ be a c-secure with ε_{in}-error family of binary (l, q)-codes, as in Definition 2.12. For every code C_s^{in}, let ϕ_s denote a bijective mapping $\phi_s : Q \to C_s^{\text{in}}$. Also, let $(s_{t1}, \ldots, s_{tn})$ be the vector indexed by t in S^n under an arbitrary total-order relation, where $1 \leq i \leq |S|^n$. Denote by C_t the code constructed in the following way:

$$C_t \stackrel{\text{def}}{=} \{\Phi_t(\mathbf{w}) : \mathbf{w} \in C_{\text{out}}\}, \tag{3.15}$$

where

$$\Phi_t(\mathbf{w}) \stackrel{\text{def}}{=} (\phi_{s_{t1}}(w_1), \ldots, \phi_{s_{tn}}(w_n)).$$

The set $\mathcal{C} = \{C_t\}_{t \in T}$, with $T = \{1, \ldots, |S|^n\}$, constitutes the concatenated family.

To use the family $\mathcal{C} = \{C_t\}_{t \in T}$ from the construction above, the distributor chooses the code C_t, where $1 \leq t \leq |T|$ is chosen with probability $\pi(t) = |T|^{-1}$. Note that this is equivalent to obtain a vector of keys $(s_1, \ldots, s_n)$, where each key is chosen independently and uniformly from S, and with this vector construct the code C_t as in (3.15). Recall that the actual value of t is kept secret. The set of keys S, the family $\mathcal{C}_{\text{in}}$, the mappings ϕ_s and the code C_{out} are publicly known. The distributor assigns to each user a codeword from C_t. Moreover, since the number of codewords in C_t and the number of codewords in C_{out} coincide, the distributor can also identify users by codewords of C_{out}.

It is worth noting that for the inner family of codes in Construction 3.14, we do not attach ourselves to any specific fingerprinting family proposal. Instead, we let $\mathcal{C}_{\text{in}}$ be any c-secure with ε_{in}-error family of binary (l, q)-codes, as in Definition 2.12.

Given a descendant

$$\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n}) \in \text{desc}_c(C_t),$$

Algorithm 3.4 Concatenated Tracing Algorithm 1

For notational simplicity, assume that $(s_1, \ldots, s_n) = (s_{t1}, \ldots, s_{tn})$.

Input: A concatenated code C_t from Construction 3.14, and a descendant $\mathbf{z} \in \mathrm{desc}_c(C_t)$,

$$\mathbf{z} = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n}).$$

Output: A codeword of C_t.

1) Let A_s^{in} denote the identification algorithm for the inner code C_s^{in}. Use the secret key s_i to decode each block $\mathbf{z}_i = (z_{i1}, \ldots, z_{il})$ of the descendant $\mathbf{z}$ by running the identification algorithm $A_{s_i}^{\mathrm{in}}(\mathbf{z}_i)$. According to Definition 2.12, we obtain at most c codewords from $C_{s_i}^{\mathrm{in}}$.

2) For $1 \leq i \leq n$, use the inverse mapping

$$\phi_{s_i}^{-1} : C_{s_i}^{\mathrm{in}} \to Q$$

to obtain a set $\mathcal{Z}_i$ of at most c symbols from Q. We pick at random one of these symbols, say symbol $Z_i \in \mathcal{Z}_i$.

3) Construct the word

$$\mathbf{Z} := (Z_1, \ldots, Z_n) \in Q^n.$$

4) Compute the codeword $\hat{\mathbf{w}} \in C_{\mathrm{out}}$ such that

$$d(\hat{\mathbf{w}}, \mathbf{Z}) = \min_{\mathbf{w} \in C_{\mathrm{out}}} d(\mathbf{w}, \mathbf{Z}),$$

and output $\hat{\mathbf{u}} = \Phi_t(\hat{\mathbf{w}}) \in C_t$.

created by a traitor coalition of size at most c, it is clear that in order to identify the traitors we first need to perform identification in each inner code and from the obtained result perform identification in the outer code. This is made precise in Algorithm 3.4, which corresponds to the identification algorithm of code C_t. Below, we will see what conditions the family of codes needs to satisfy so that the output of the algorithm is a traitor codeword with high probability.

We are now in the position to state the following theorem.

Theorem 3.15. Let C_{out} be an (n, M)-code over a q-ary alphabet Q, and let $\mathcal{C}_{\text{in}} = \{C_s^{\text{in}}\}_{s\in S}$ be a c-secure with ε_{in}-error family of binary (l, q)-codes, as in Definition 2.12. Let $\mathcal{C} = \{C_t\}_{t\in T}$ be the family of concatenated codes from Construction 3.14 with outer code C_{out}, the family of inner codes $\mathcal{C}_{\text{in}}$, the mappings ϕ_s, the set of keys T, and $\pi(t) = |T|^{-1}$. For any σ, where $\varepsilon_{\text{in}} < \sigma < 1/(c+1)$, the family of concatenated codes $\mathcal{C} = \{C_t\}_{t\in T}$ together with Algorithm 3.4 is a c-secure family of binary codes with exponentially small error, $\varepsilon = \exp(-\Omega(n))$, if

$$d(C_{\text{out}}) > n - \frac{n(1-\sigma(c+1))}{c^2}.$$

Proof. Let C_t be the code chosen by the distributor. For convention, assume that $(s_1, \ldots, s_n) = (s_{t1}, \ldots, s_{tn}) \subseteq S^n$ is the set of keys of the inner family corresponding to the chosen key $t \in T$. Note that this choice of t is equivalent to select the vector $(s_1, \ldots, s_n)$ at random, where each entry is chosen uniformly and independently from S. This choice is kept secret, but the rest of parameters from the family are public.

Let $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C_t$ be a c-coalition, and also let the subset of their corresponding outer codewords be $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^c\} \in C_{\text{out}}$. That is, the codewords $\mathbf{u}^j = (\phi_{s_1}(w_1^j), \ldots, \phi_{s_n}(w_n^j)) \in C_t$, and $\mathbf{w}^j = (w_1^j, \ldots, w_n^j) \in C_{\text{out}}$, for $1 \le j \le c$. Also, let

$$\mathbf{z} = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n})$$

be a descendant created by coalition U. We use Algorithm 3.4 to identify traitors. By decoding each block $\mathbf{z}_i = (z_{i1}, \ldots, z_{il})$, following Steps 1) and 2) of the algorithm, we obtain a symbol $Z_i \in Q$. Recall from (2.2) that the projection of W on the ith

position is defined as the set of the symbols of the code alphabet in that position,

$$P_i(W) \stackrel{\text{def}}{=} \{w_i^1, \ldots, w_i^c\}.$$

Hence, according to Definition 2.12, Z_i matches one of the outer traitor codewords, i.e., $Z_i \in P_i(W)$, with probability at least $1 - \varepsilon_{\text{in}}$.

For any given descendant $\mathbf{z}$ the errors in the decoding of each block $\mathbf{z}_i$ are independent. To see this, we recall that the keys $s_1, \ldots, s_n$ are chosen independently and uniformly at random. In other words, the inner codes $C_{s_i}^{\text{in}}$ together with their associated mappings ϕ_{s_i} are chosen independently and uniformly from the the family C_{in}. Then, it is clear that the errors made in each identification algorithm of the inner code $A_{s_i}^{\text{in}}$ are independent.

Now, let X be the total number of errors made by the identification algorithm of the inner code. Hence, the r.v. X can be viewed as the sum of n independent indicator r.v.'s with probability of success $\leq \varepsilon_{\text{in}}$ each. We can bound $\Pr\{X \geq x\}$, by comparing X with an appropriate binomial r.v., of parameters n and ε_{in}. Then, using (2.10),

$$\Pr\{X \geq n\sigma\} \leq 2^{-nD(\sigma \| \varepsilon_{\text{in}})} \leq e^{-2n(\sigma - \varepsilon_{\text{in}})^2}. \tag{3.16}$$

Thus, after decoding the inner codes, we recover a word over the alphabet of the outer code, $\mathbf{Z} = (Z_1, \ldots, Z_n) \in Q^n$, with the property

$$\Pr\{|\{Z_i : Z_i \in P_i(W)\}| > n - n\sigma\} \geq 1 - e^{-2n(\sigma - \varepsilon_{\text{in}})^2}. \tag{3.17}$$

That is, with error probability $\varepsilon \leq e^{-2n(\sigma - \varepsilon_{\text{in}})^2}$, the number of incorrectly decoded positions in $\mathbf{Z}$ is at most $n\sigma$. This means that, with exponentially small error probability ε, there exists some coalition codeword $\hat{\mathbf{u}} = \Phi_t(\hat{\mathbf{w}}) \in U$ for some $\hat{\mathbf{w}} \in W$ such that the similitude with $\mathbf{Z}$ satisfies

$$\mathrm{s}(\hat{\mathbf{w}}, \mathbf{Z}) \geq n\frac{1 - \sigma}{c}. \tag{3.18}$$

Note that since $\varepsilon \leq e^{-2n(\sigma-\varepsilon_{\text{in}})^2}$, then for reasonable values of n we have

$$\varepsilon < \varepsilon_{\text{in}}.$$

Implied by the condition in the minimum distance of C_{out}, any two codewords $\mathbf{v}, \mathbf{w} \in C_{\text{out}}$ satisfy

$$\mathrm{s}(\mathbf{v}, \mathbf{w}) < \frac{n(1-\sigma(c+1))}{c^2}.$$

Recalling that with high probability $n\sigma$ is an upper bound on the number of positions such that $Z_j \notin P_i(W) = \{w_j^1, \ldots, w_j^c\}$, for any corresponding innocent codeword from the outer code $\mathbf{v} \in C_{\text{out}} \setminus W$, we have

$$\begin{aligned} \mathrm{s}(\mathbf{v}, \mathbf{Z}) &\leq n\sigma + \sum_{j=1}^{c} \mathrm{s}(\mathbf{v}, \mathbf{w}^j) \\ &< n\sigma + c\frac{n(1-\sigma(c+1))}{c^2} = n\frac{1-\sigma}{c}. \end{aligned} \tag{3.19}$$

Putting together (3.18) and (3.19), with error probability less than $e^{-2n(\sigma-\varepsilon_{\text{in}})^2}$, the closest codeword $\hat{\mathbf{w}} \in C_{\text{out}}$ to $\mathbf{Z}$ is the outer codeword corresponding to a traitor codeword, i.e., $\hat{\mathbf{u}} = \Phi_t(\hat{\mathbf{w}}) \in U$. This is precisely the output of Algorithm 3.4. □

A related result has been obtained independently in [52, 53].

3.4.1 Efficient Identification of Traitors

We have just shown that there exists a family of binary fingerprinting codes, based on a concatenated construction, which together with Algorithm 3.4 can achieve identification of traitors with arbitrarily small error. Now, we will show how the complexity of the identification process in a concatenated code can be reduced using the Kötter-Vardy algorithm when Reed-Solomon codes are used as outer codes.

Similarly to the Algorithm 3.4, the identification process of concatenated codes is usually performed in two steps. In the first step, every inner code is decoded obtaining a word of symbols from the outer code alphabet. Then, this word is decoded using a decoding algorithm designed for the outer code.

As stated in Remark 2.4, in a collusion attack, the output of the identification algorithm of each inner code need not be a single symbol from the outer code alphabet. It can also be a set of multiple symbols. All the (possible) multiple outputs are considered to have the same reliability.

We will consider henceforth that the outer code C_{out} from Construction 3.14 is an $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$. In this case, through the use of the reliability matrix, the Kötter-Vardy algorithm provides a natural way to deal with all the information delivered by the inner identification process. Using the same notation as above, we reflect this situation in Algorithm 3.5.

As in the proof of Theorem 3.15, let $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C_t$ denote a c-coalition, and let $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^c\} \subseteq C_{\text{out}}$ be the subset of their corresponding outer codewords.

In Step 2) of the algorithm we recover a set of symbols $\mathcal{Z}_i$ such that, from Definition 2.12, it is a nonempty subset satisfying

$$\mathcal{Z}_i \subseteq P_i(W) = \{w_i^1, \ldots, w_i^c\}$$

with probability $\geq 1 - \varepsilon_{\text{in}}$, for $1 \leq i \leq n$.

If no errors have been made in the inner identification process, i.e., $\mathcal{Z}_i$ is nonempty and $\mathcal{Z}_i \subseteq P_i(W)$ for $1 \leq i \leq n$, then there is a traitor codeword $\mathbf{w} \in W$ such that the similitude, as defined in (2.1), satisfies $\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n/c$. However, following the same reasoning used to obtain (3.16) and (3.17), it can only be guaranteed with high probability that the inner identification process has made less than $n\sigma$ errors. Hence the previous condition needs to be reformulated as

$$\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c}, \tag{3.20}$$

for some $\mathbf{w} \in W$. It will only remain to show that such $\mathbf{w}$ will be returned by the Kötter-Vardy algorithm.

Let us assume that $|\mathcal{Z}_i| = c$ for all $1 \leq i \leq n$. It is easy to see that, in particular for $\sigma < 1/(c+1)$, this is indeed a worst-case situation in the analysis below. Namely, the one which minimizes the difference between the left-hand and the right-hand

Algorithm 3.5 Concatenated Tracing Algorithm 2

Initial ordering of the elements of $\mathbb{F}_q$: $\alpha_1, \alpha_2, \ldots, \alpha_q$. For notational simplicity, assume that $(s_1, \ldots, s_n) = (s_{t1}, \ldots, s_{tn})$.

Input: A concatenated code C_t from Construction 3.14, using an $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$ as outer code, a bound σ for the inner code error probability such that $\varepsilon_{\text{in}} < \sigma < 1/(c+1)$, and a descendant $\mathbf{z} \in \text{desc}_c(C_t)$,

$$\mathbf{z} = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n}).$$

Output: A subset of codewords of C_t.

1) Let A_s^{in} denote the identification algorithm for the inner code C_s^{in}. Use the secret key s_i to decode each block $\mathbf{z}_i = (z_{i1}, \ldots, z_{il})$ of the descendant $\mathbf{z}$ by running the identification algorithm $A_{s_i}^{\text{in}}(\mathbf{z}_i)$. According to Definition 2.12, we obtain at most c codewords from $C_{s_i}^{\text{in}}$.

2) For $1 \leq i \leq n$, use the inverse mapping

$$\phi_{s_i}^{-1} : C_{s_i}^{\text{in}} \to \mathbb{F}_q$$

to obtain a set $\mathcal{Z}_i$ of at most c symbols from $\mathbb{F}_q$.

3) Construct the set vector

$$\boldsymbol{\mathcal{Z}} := (\mathcal{Z}_1, \ldots, \mathcal{Z}_n).$$

4) Set $\xi := \sigma$ and compute the q-by-n reliability matrix $\mathcal{R} = (r_{ji})$, where for $1 \leq j \leq q$ and $1 \leq i \leq n$,

$$r_{ji} := \begin{cases} \dfrac{1-\xi}{|\mathcal{Z}_i|} & \text{if } \alpha_j \in \mathcal{Z}_i, \\ \dfrac{\xi}{q - |\mathcal{Z}_i|} & \text{otherwise.} \end{cases}$$

5) Plug the matrix $\mathcal{R}$ into the Kötter-Vardy algorithm and for every codeword $\mathbf{w} = (w_1, \ldots, w_n) \in C_{\text{out}}$ in the output list, compute the similitude $\text{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}})$, according to (2.1).

6) Output the set $L := \{\mathbf{u}^1, \ldots, \mathbf{u}^s\}$, consisting of all codewords $\mathbf{u} = \Phi_t(\mathbf{w}) \in C_t$, such that

$$\text{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c},$$

for some codeword $\mathbf{w}$ obtained in Step 5).

sides in the condition for successful decoding of the Kötter-Vardy algorithm stated in Theorem 3.5,

$$\frac{\langle \mathcal{R}, [\mathbf{w}] \rangle}{\sqrt{\langle \mathcal{R}, \mathcal{R} \rangle}} \geq \sqrt{k-1} + o(1). \tag{3.21}$$

For a codeword $\mathbf{w} \in W$ satisfying (3.20), we have

$$\langle \mathcal{R}, [\mathbf{w}] \rangle = (n - n\sigma)\frac{1-\xi}{c} + n\sigma\frac{\xi}{q-c},$$

where $\mathcal{R}$ is the reliability matrix constructed in Step 4), and since

$$\sqrt{\langle \mathcal{R}, \mathcal{R} \rangle} = \sqrt{\frac{n}{c}(1-\xi)^2 + \frac{n}{q-c}\xi^2},$$

then $\mathbf{w}$ will appear in the output list of the Kötter-Vardy algorithm if

$$\frac{(n - n\sigma)\frac{1-\xi}{c} + n\sigma\frac{\xi}{q-c}}{\sqrt{\frac{n}{c}(1-\xi)^2 + \frac{n}{q-c}\xi^2}} > \sqrt{\frac{n(1-\sigma(c+1))}{c^2}}. \tag{3.22}$$

The reason for ξ in the algorithm is to take into account the fact that the inner code identification algorithm has a certain error probability. In this way, if in a given position an error is made, then in this position we will still have some "contribution" of the traitors.

The left-hand side of (3.22) is maximized by taking $\xi = \sigma$. This is intuitively very satisfactory. It says that, in the setup of the reliability matrix $\mathcal{R}$, the effect of the errors of the inner binary fingerprinting code has to be considered. Moreover, this effect has to be "spread" equally between all symbols that do not appear in the list returned by the inner identification process. From Theorem 3.15, note that σ is an upper bound on the error probability of each inner code, and represents a threshold that allows us to "differentiate" parents (traitors) from non-parents.

Note that the analysis above implies that, with error probability $\varepsilon = \exp(-\Omega(n))$, no innocent codeword will be accused, and every codeword $\mathbf{u} = \Phi_t(\mathbf{w})$ such that $\mathbf{w} \in C_{\text{out}}$ satisfies condition (3.20) can be accused as a traitor. This is in fact a list-decoding algorithm that returns all codewords that satisfy (3.20).

Finally note that, if done by brute force, the identification process of each inner code is of complexity $O(ln)$, as in [13, 14, 18]. This means a decoding complexity of $O(ln^2)$ for all the entire inner decoding, where n is the outer code length. Since the Kötter-Vardy algorithm is the core of Algorithm 3.5, and it is a polynomial-time algorithm in the code length [44], we conclude that the identification process is also accomplished in polynomial time in the total code length.

Thus, we have proved the following proposition.

Proposition 3.16. Let C_{out} be an $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$, and let $\mathcal{C}_{\text{in}} = \{C_s^{\text{in}}\}_{s\in S}$ be a c-secure with ε_{in}-error family of binary (l, q)-codes, as in Definition 2.12. Let $\mathcal{C} = \{C_t\}_{t\in T}$ be the family of concatenated codes from Construction 3.14 with outer code C_{out}, the family of inner codes $\mathcal{C}_{\text{in}}$, the mappings ϕ_s, the set of keys T, and $\pi(t) = |T|^{-1}$. For any σ, where $\varepsilon_{\text{in}} < \sigma < 1/(c + 1)$, the family of concatenated codes $\mathcal{C} = \{C_t\}_{t\in T}$ together with Algorithm 3.5 is a c-secure family of binary codes with exponentially small error, $\varepsilon = \exp(-\Omega(n))$, if

$$d(C_{\text{out}}) > n - \frac{n(1 - \sigma(c + 1))}{c^2}.$$

Moreover, the identification process is executed in polynomial time in the code length, and its capacity is maximized by using as the input to the Kötter-Vardy algorithm a reliability matrix $\mathcal{R}$ that has $\leq c$ entries of value $\geq (1 - \sigma)/c$ and $\geq c$ entries of value $\leq \sigma/(q - c)$ in each column.

3.4.2 Suboptimal Setup of the Reliability Matrix

In the situation discussed above, we argued that the reason for ξ was to take into account the errors made by the identification process of the inner codes. At that point, the reader might have thought about what would happen if one had decided to ignore the fact that the inner family of codes C_{in} has a probability of error. This

would mean constructing the reliability matrix $\mathcal{R} = (r_{ji})$ as

$$r_{ji} = \begin{cases} \dfrac{1}{|\mathcal{Z}_i|} & \text{if } \alpha_j \in \mathcal{Z}_i, \\ 0 & \text{otherwise.} \end{cases}$$

Again, we assume that the number of errors made by the identification process of the inner codes are at most $n\sigma$ with high probability, and the worst-case situation where $|\mathcal{Z}_i| = c$ for $1 \leq i \leq n$. Then, for an outer codeword $\mathbf{w} \in W$, we have

$$\langle \mathcal{R}, [\mathbf{w}] \rangle = \frac{n - n\sigma}{c}.$$

Since now

$$\sqrt{\langle \mathcal{R}, \mathcal{R} \rangle} = \sqrt{\frac{n}{c}},$$

then, according to (3.21), the codeword $\mathbf{w}$ will appear in the output list of the Kötter-Vardy algorithm if

$$\frac{(n - n\sigma)/c}{\sqrt{n/c}} > \sqrt{\frac{n(1 - \sigma(c+1))}{c^2}}.$$

This is the same as

$$1 - \sigma > \sqrt{\frac{1 - \sigma(c+1)}{c}},$$

which for $\sigma < 1/(c+1)$ is always satisfied.

This means that even for a suboptimal setup of the reliability matrix, the Kötter-Vardy algorithm will output a codeword from the traitor coalition. In a way, this is a surprising result. Recall from Section 3.2 that to find the TA-parents in the TA Tracing Algorithm we had to push the Kötter-Vardy algorithm almost "to the edge." This was due to the fact that every column of the reliability matrix only contained information from a single parent. On the other hand, here we are able to exploit the full power of the Kötter-Vardy algorithm. This is because, whenever it is possible, each column of the reliability matrix contains information from all parents. Somehow, it looks as if the Kötter-Vardy algorithm is tailor made for these concatenated constructions.

3.5 Conclusion

As noted in [36,37], tracing traitors is a worthwhile addition to a system provided that the associated identification algorithms add sufficiently little cost. In this chapter we have shown the benefits of using the Kötter-Vardy soft-decision decoding algorithm in the identification process when Reed-Solomon codes with tracing capabilities are used.

For TA Reed-Solomon codes, on one hand, we give conditions for unambiguous traitor identification. On the other hand, we show how the flexibility of the Kötter-Vardy algorithm allows the reuse of information obtained in each loop of an iterative process, in which the identification of traitors is based on the previously identified ones. The use of feedback information from previous iterations of the algorithm improves the task, allowing it to run in polynomial time in the code length, rather than in the code size. We also discuss upper bounds of the needed cost in the Kötter-Vardy algorithm so that at least one TA-parent always appears in the output list.

Moreover, we have also extended the work of [36, 37]. Again departing from the Kötter-Vardy algorithm, for a c-IPP Reed-Solomon code, given a descendant we have presented a method to obtain all possible coalitions that are able to generate it. The use of the soft-decision decoding routine allows us to reduce the execution time, which in the general case is upper-bounded by $O\binom{M}{c}$, where M is the total number of codewords.

Finally, we have shown concatenated constructions of binary fingerprinting codes based on Reed-Solomon outer codes. The constructions have exponentially small error probability in the outer code length, and polynomial decoding time in the total code length. We use the Kötter-Vardy soft-decision decoding algorithm in the outer code identification process. It is noticeable that even a sub-optimal setup of the reliability matrix achieves the same purposes than the matrix defined for the optimal case and with equivalent computational complexity.

The contents of this chapter have been published in [3], and also in the joint works [6] and [7].

Chapter 4

Almost Separating and Almost Secure Frameproof Codes

Separating codes were introduced by Friedman et al. [20] more than 40 years ago. A separating code is a very natural combinatorial object that has found application in many areas. Fields such as automata synthesis, technical diagnosis, construction of hash functions and traitor-tracing schemes have benefited from codes with the separating property.

As commented in Section 2.1, separating codes have been subsequently investigated by many authors, e.g. in [21,22,23,24,25,26]. Nontrivial lower and upper bounds have been derived and relationships with similar notions have been established. See for instance the surveys [21, 25].

Recently, in connection with digital fingerprinting codes, a great deal of attention has been paid to separating codes. In this new area of application, separating codes have been rediscovered under the names of frameproof and secure frameproof codes [13, 14, 27, 28].

The main note of this chapter is the fact that relaxing the definitions of separating and secure frameproof codes, by demanding that these properties (separating and secure frameproofness) hold with high probability, will bring us two different notions. We call these two new notions *almost separating* and *almost secure frameproof* property. As it will be shown, allowing a code that the separating property holds with high

probability, as opposed to absolute separation, allows us to obtain codes with better rates. Namely, we show existence bounds for almost separating and almost secure frameproof codes that are better than the current existence bounds for separating codes.

This chapter is organized as follows. In Section 4.1 we introduce the topic and present some previous results. In Section 4.2 and Section 4.3, we obtain lower bounds on the rate of the new codes introduced. Next, in Section 4.4 we compare the obtained results with the current known state of the art. Our motivation for studying separating codes is their application to fingerprinting schemes. In Section 4.5, we construct a family of fingerprinting codes with small error using almost separating and almost secure frameproof codes. Finally, the conclusions are drawn in Section 4.6.

4.1 Separating and Secure Frameproof Codes Revisited

Let C be an (n, M)-code. For a pair of (disjoint) subsets $U, V \subseteq C$, using the notation from (2.2), we say that a position i is *separating* if

$$P_i(U) \cap P_i(V) = \emptyset.$$

A pair of c-subsets U, V are called *separated* if there exists a separating position $1 \leq i \leq n$ for them. Moreover, we say that a c-subset U is *separated* if U is separated from every other disjoint c-subset $V \subseteq C$.

Now, Definition 2.7 can be restated, and a code C can be defined as (c, c)-separating if every pair of disjoint c-subsets $U, V \subseteq C$ are separated. Equivalently, a code is (c, c)-separating if every c-subset $U \subseteq C$ is separated. We have the following definitions.

Definition 4.1. A code C is *c-frameproof* if every set $U \subseteq C$ with $|U| \leq c$ satisfies $\text{desc}(U) \cap C = U$.

Definition 4.2. A code C is *c-secure frameproof* if for any $U, V \subseteq C$ with $|U| \leq c$, $|V| \leq c$ and $U \cap V = \emptyset$, then $\mathrm{desc}(U) \cap \mathrm{desc}(V) = \emptyset$.

The concepts of frameproof and secure frameproof codes were introduced in [13, 14, 27, 28]. It is easy to see, and it was clearly noticed, e.g. in [15], that a c-frameproof code is the same as a $(c, 1)$-separating code, and that a c-secure frameproof code is the same as a (c, c)-separating code.

Let $R_q^{\mathrm{sep}}(n, c, c')$ denote the rate of an optimal (i.e., maximal) (c, c')-separating code of length n over a q-ary alphabet Q,

$$R_q^{\mathrm{sep}}(n, c, c') \stackrel{\mathrm{def}}{=} \max_{\substack{C \subseteq Q^n \text{ s.t. } C \text{ is} \\ (c,c')\text{-separating}}} R(C).$$

Also, consider the corresponding asymptotical rates

$$\underline{R}_q^{\mathrm{sep}}(c, c') \stackrel{\mathrm{def}}{=} \liminf_{n\to\infty} R_q^{\mathrm{sep}}(n, c, c'), \qquad \overline{R}_q^{\mathrm{sep}}(c, c') \stackrel{\mathrm{def}}{=} \limsup_{n\to\infty} R_q^{\mathrm{sep}}(n, c, c').$$

Lower bounds on $(2, 2)$-separating codes were studied in [20, 22]. For binary separating codes there are some important, well-known results that are worth mentioning. For example, from [21, 22] we have $\underline{R}_2^{\mathrm{sep}}(2, 2) \geq 1 - \log_2(7/8) = 0.0642$, which also holds for linear codes [22]. Also, for the general case, it was shown in [15] that

$$\underline{R}_2^{\mathrm{sep}}(c, c') \geq -\frac{\log_2(1 - 2^{-c-c'+1})}{c + c' - 1}. \tag{4.1}$$

Regarding the upper bounds, in [21, 24] it was shown that $\overline{R}_2^{\mathrm{sep}}(2, 2) < 0.2835$ for arbitrary codes, and in [21] that $\overline{R}_2^{\mathrm{sep}}(2, 2) < 0.108$ for linear codes.

In the following sections of this chapter, and unless otherwise stated, all random (n, M)-codes are considered to be chosen with uniform probability among the ensemble of all (n, M)-codes over a certain alphabet Q. That is, we generate M vectors of length n, where each entry is uniformly and independently chosen from Q.

4.2 Separating and Almost Separating Codes over q-ary Alphabets

We start the study of separating and almost separating codes by obtaining lower bounds for separating codes over arbitrary alphabets. This will allow us to compare these results with the concepts of almost separating and almost secure frameproof codes that we are introducing. We will use a standard probabilistic argument to obtain a generalization of (4.1).

4.2.1 Lower Bounds for q-ary Separating Codes

Lemma 4.3. Let $v(j;q,c)$ be the pmf, evaluated at j, of an r.v. that counts the number of different symbols of a q-ary vector of length c chosen uniformly at random. Then,

$$v(j;q,c) \stackrel{\text{def}}{=} \frac{q^{\underline{j}}}{q^c} \left\{ {c \atop j} \right\}, \qquad 1 \leq j \leq \min\{q,c\}, \tag{4.2}$$

where $q^{\underline{j}}$ denotes the falling factorial and $\left\{ {c \atop j} \right\}$ denotes the Stirling number of the second kind.

Proof. A subset of size c can be partitioned into j nonempty subsets in $\left\{ {c \atop j} \right\}$ different ways. For each such partition there are $q(q-1)\cdots(q-j+1) = q^{\underline{j}}$ possible assignments using j different elements from Q. The product of these two terms gives the number of q-ary vectors of length c that contain exactly j different symbols. The proof follows after dividing by the total number of vectors. □

For notational simplicity, we will sometimes suppress the parameters q, c from $v(j;q,c)$, and we will refer to this pmf simply as $v(j)$. Fortunately these parameters will be clear from the context. Also, we will often omit the range of the support of v in the summation indices, which will always be understood as above. In fact, one could chose either parameter (q or c) arbitrarily as the upper limit in the range of $v(j;q,c)$. By definition $v(j;q,c)$ will evaluate to 0 for $j \neq 1,\ldots,\min\{q,c\}$.

Lemma 4.4. Let $p^{\text{disj.}}_{q,c,c'}$ be the probability that two q-ary vectors of lengths c and c', respectively, chosen uniformly and independently at random are disjoint (i.e., have no common element). We have

$$p^{\text{disj.}}_{q,c,c'} = \sum_j (1 - j/q)^{c'}\, v(j; q, c),$$

where j ranges over the support of the pmf v, defined in (4.2).

Proof. Let $\mathbf{a} = (a_1, \ldots, a_c)$ and $\mathbf{b} = (b_1, \ldots, b_{c'})$ be two random vectors, of length c and c', respectively, and let X be the r.v. that counts the number of different symbols in $\mathbf{a}$. The probability that $\mathbf{a}$ and $\mathbf{b}$ are disjoint, i.e., $\{a_1, \ldots, a_c\} \cap \{b_1, \ldots, b_{c'}\} = \emptyset$, can be computed as

$$p^{\text{disj.}}_{q,c,c'} = \sum_j \Pr\{\mathbf{a} \text{ and } \mathbf{b} \text{ disjoint} \,|\, X = j\} \Pr\{X = j\}.$$

Clearly, $\Pr\{X = j\} = v(j; q, c)$. Also, since $\mathbf{b}$ has c' elements independently chosen from $\mathbf{a}$, we have $\Pr\{\mathbf{a} \text{ and } \mathbf{b} \text{ disjoint} \,|\, X = j\} = (1 - j/q)^{c'}$. □

Note that, given two c-subsets U, V of a random q-ary (n, M)-code C, the probability that a position i is separating, i.e., $P_i(U) \cap P_i(V) = \emptyset$ is precisely $p^{\text{disj.}}_{q,c,c'}$. Using this fact, combined with the probabilistic argument borrowed from [15, Proposition 3.4], the following result follows easily. We provide the proof below for completeness.

Corollary 4.5. There exist q-ary (c, c')-separating codes of asymptotical rate satisfying

$$\underline{R}^{\text{sep}}_q(c, c') \geq -\frac{\log_q(1 - p^{\text{disj.}}_{q,c,c'})}{c + c' - 1}.$$

Proof. Let C be a random q-ary (n, M)-code, and let E be the expected number of "bad" pairs U, V of subsets with $|U| = c$ and $|V| = c'$, i.e., pairs that are not separated. If $E < M/2$, then a q-ary $(n, M/2)$-code with the (c, c')-separating property exists, since by removing one codeword from each bad pair, the remaining codewords yield a (c, c')-separating code. The probability that a pair U, V of such subsets is not

separated is $(1 - p_{q,c,c'}^{\text{disj.}})^n$. Hence, we have

$$E \leq \binom{M}{c}\binom{M-c}{c'}(1 - p_{q,c,c'}^{\text{disj.}})^n < \frac{M^{c+c'}}{c!\,c'!}(1 - p_{q,c,c'}^{\text{disj.}})^n.$$

Observe that taking $M = \left(\frac{c!c'!}{2}(1 - p_{q,c,c'}^{\text{disj.}})^{-n}\right)^{1/(c+c'-1)}$, we have $E < M/2$. Finally, since $(\frac{c!c'!}{2})^{1/(c+c'-1)} \geq 1$, we can disregard the logarithm of this term in the lower bound on the code rate. □

4.2.2 Lower Bounds for Almost Separating Codes

The separating property imposes a very strict combinatorial restriction to the code, namely that every pair of c-subsets is separated. One could obtain codes with better rates by relaxing this condition, and asking for codes where it is satisfied with high probability, rather than in all cases. In this section we propose one possible way of relaxing the separating condition.

Recall again that for a code C, a c-subset $U \subseteq C$ is separated if U is separated from every other disjoint c-subset $V \subseteq C$. Now, we have the following definition.

Definition 4.6. A code C is *ε-almost (c, c)-separating* if the ratio of c-subsets that are separated is at least $1 - \varepsilon$.

A sequence of codes $\mathcal{C} = (C_i)_{i\geq 1}$ of growing length n_i is an *asymptotically almost (c, c)-separating family* if every code C_i is ε_i-almost (c, c)-separating and $\lim_{i\to\infty} \varepsilon_i = 0$.

We also define the asymptotical rate of a sequence $\mathcal{C} = (C_i)_{i\geq 1}$ as

$$R(\mathcal{C}) = \liminf_{i\to\infty} R(C_i). \tag{4.3}$$

We are interested in estimating the maximal possible asymptotical rate, denoted $R_q^{\text{sep}^*}(c)$, among all asymptotically almost (c, c)-separating families.

To derive lower bounds, we make use of a restricted version of strongly typical subsets of codewords [54]. That is, subsets of codewords that appear with high probability in a random code.

Let C be a q-ary (n, M)-code, and let $U \subseteq C$ be a c-subset. We say that a position i is *j-valued* if its projection $P_i(U)$ contains exactly j different symbols from the code alphabet. We denote $N(j;U)$, for $1 \leq j \leq \min\{q, c\}$, the number of positions i that are j-valued. For example, if $Q = \{0, 1, 2\}$ and

$$\begin{aligned} U = \{\ & (2,1,0,0,2,0,0,1,0,0,0,2,1,0,2), \\ & (1,1,1,1,1,0,0,2,1,0,0,0,0,0,2), \\ & (1,2,2,0,1,0,1,0,0,0,1,1,0,2,0), \\ & (2,1,1,1,1,1,0,2,0,0,2,1,1,1,2)\ \}, \end{aligned}$$

then $N(1;U) = 1$, $N(2;U) = 9$, $N(3;U) = 5$ and $N(j;U) = 0$ otherwise. Note that for a c-subset U uniformly chosen from a random (n, M)-code, the empirical distribution $n^{-1}N(j;U)$ satisfies

$$E[n^{-1}N(j;U)] = v(j), \qquad 1 \leq j \leq \min\{q, c\}.$$

We say that the c-subset U is *δ-typical* if the empirical distribution of the number of j-valued positions, i.e., $n^{-1}N(j;U)$, is "close" to the expected value $v(j)$ of a c-subset in a random code. Namely, for $0 < \delta \leq 1$,

$$|n^{-1}N(j;U) - v(j)| < \delta, \qquad 1 \leq j \leq \min\{q, c\}.$$

Also, we denote by $A_\delta^{(n)}(q, c)$ the set of δ-typical c-subsets of C,

$$A_\delta^{(n)}(q, c) \stackrel{\text{def}}{=} \{U \subseteq C : |U| = c \text{ and } U \text{ is } \delta\text{-typical}\}. \tag{4.4}$$

Note that each $N(j;U)$ can be regarded as a binomial r.v. of parameters n and $v(j)$. Then, combining the union bound with (2.10) and (2.11), it is not difficult to see that the probability that a randomly and uniformly chosen c-subset U is not contained in the typical set satisfies

$$\Pr\{U \notin A_\delta^{(n)}(q, c)\} \leq \sum_j 2^{-nD(v(j)-\delta\|v(j))} + 2^{-nD(v(j)+\delta\|v(j))} \leq 2q\, e^{-2n\delta^2}. \tag{4.5}$$

With these results in mind, we are ready to derive a lower bound for q-ary almost separating codes.

Theorem 4.7. For the maximal possible asymptotical rate $R_q^{\mathrm{sep}^*}(c)$ among all asymptotically almost (c,c)-separating families of q-ary codes we have

$$R_q^{\mathrm{sep}^*}(c) \geq -\frac{1}{c}\sum_j \log_q(1-(1-j/q)^c)\, v(j).$$

Proof. Consider a random q-ary (n,M)-code C. For a given c-subset $U \subseteq C$ there are exactly $N(j;U)$ j-valued positions. For each such position i, the probability that another random c-subset V satisfies $P_i(U) \cap P_i(V) = \emptyset$ equals $(1-j/q)^c$. Thus,

$$\Pr\{U \text{ and } V \text{ are not separated}\} = \prod_j (1-(1-j/q)^c)^{N(j;U)}.$$

Let U be a typical c-subset as defined in (4.4). Using (4.5), the probability ε that U is not separated satisfies

$$\begin{aligned}
\varepsilon &= \Pr\{U \text{ is not separated} \,|\, U \text{ is typical}\}\Pr\{U \text{ is typical}\} + \\
&\qquad \Pr\{U \text{ is not separated} \,|\, U \text{ is not typical}\}\Pr\{U \text{ is not typical}\} \\
&\leq \Pr\{U \text{ is not separated} \,|\, U \text{ is typical}\} + \Pr\{U \text{ is not typical}\} \\
&\leq \binom{M-c}{c} \prod_j (1-(1-j/q)^c)^{n(v(j)-\delta)} + 2q\, e^{-2n\delta^2}.
\end{aligned}$$

Hence, we have

$$\lim_{n\to\infty} \frac{1}{n}\log_q \varepsilon \leq c\,R + \sum_j \log_q(1-(1-j/q)^c)\, v(j).$$

Now take a sequence of codes $\mathcal{C} = (C_i)_{i\geq 1}$ of growing length such that each (n_i, M_i)-code C_i is a random code. The probabilistic argument above shows that taking an appropriate value for δ_i, for example $\delta_i = \delta_i(n_i) = \ln n_i/\sqrt{n_i}$, we conclude

that there exists a sequence with $\lim_{i\to\infty} \varepsilon_i = 0$ for any rate

$$R < -\frac{1}{c} \sum_j \log_q(1 - (1 - j/q)^c)\, v(j),$$

which completes the proof. □

4.2.3 A Refined Lower Bound for Binary Almost Separating Codes

The particular case of binary alphabets is of great importance, since many applications of coding theory, such as automata testing or digital fingerprinting codes, rely on these alphabets. Without loss of generality, we consider the binary alphabet $Q = \{0, 1\}$.

To obtain an improvement with respect to the previous case, we need to modify our definition of typical set used above (4.4). For a c-subset U of a binary code C we define $Z(x; U)$ as the number of positions i such that $P_i(U) = \{x\}$, for $x \in Q$. That is, $Z(0; U)$ and $Z(1; U)$ count the number of all-zero and all-one positions, respectively. For example, if

$$\begin{aligned} U = \{\ & (0,1,0,0,0,0,0,0,0,0,0,0,1,1,0), \\ & (1,1,1,1,1,0,0,0,1,0,0,0,0,1,0), \\ & (1,1,0,0,1,0,1,0,0,0,1,1,0,1,0), \\ & (0,1,1,1,1,1,0,0,0,0,0,1,1,1,0)\ \}, \end{aligned}$$

then $Z(0; U) = 3$ and $Z(1; U) = 2$. Note that in a c-subset U uniformly chosen from a random (n, M)-code C we have

$$E[n^{-1} Z(x; U)] = 2^{-c}.$$

Now, for $0 < \delta \leq 1$, we define the typical set $B_\delta^{(n)}(c)$ as

$$B_\delta^{(n)}(c) \stackrel{\text{def}}{=} \{U \subseteq C : |U| = c \text{ and } |n^{-1} Z(x; U) - 2^{-c}| < \delta,\ x \in Q\}.$$

That is, $B_\delta^{(n)}(c)$ contains all the c-subsets $U \subseteq C$ such that the empirical distribution of the number of all-zero and all-one positions is "close" to the expected value in a random code. Using a similar reasoning as above, for a random c-coalition U,

$$\Pr\{U \notin B_\delta^{(n)}(c)\} \leq 4\, e^{-2n\delta^2}. \tag{4.6}$$

Now, the idea is to use the fact that if a c-subset is typical with high probability, a pair of c-subsets will also be formed by typical subsets with high probability. First, we present the following result, which we will use below.

Lemma 4.8. Let $U, V \subseteq C$ be two disjoint c-subsets of a binary code C. If $Z(0;U) = Z(1;U) = Z(0;V) = Z(1;V) = n2^{-c}$, then the probability that U and V are not separated satisfies

$$\lim_{n\to\infty} \frac{1}{n} \log_2 \Pr\{U \text{ and } V \text{ are not separated}\} \leq G(c),$$

where

$$G(c) \stackrel{\text{def}}{=} \Big((1-2p+\ell)H_2\big(\tfrac{p}{1-2p+\ell}\big) + p\, H_2\big(\tfrac{\ell}{p}\big) + (1-2p)H_2\big(\tfrac{p-\ell}{1-2p}\big) - H_2(p) - (1-p)H_2\big(\tfrac{p}{1-p}\big)\Big),$$

with $p = 2^{-c}$ and $\ell = (2p - 1 + \sqrt{8p^2 - 4p + 1})/2$.

Proof. Take two random c-subsets $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\}, V = \{\mathbf{v}^1, \ldots, \mathbf{v}^c\}$, from a binary code, satisfying the conditions stated above. Define X_0 as the r.v. that counts the number of nonseparating positions i such that $P_i(U) = \{0\}$. That is, in X_0 positions i we have $u_i^1 = \cdots = u_i^c = 0$, and least one $\mathbf{v} \in V$ with $v_i = 0$. Similarly, let X_1 be the r.v. that counts the number of nonseparating positions i such that $P_i(U) = \{1\}$.

Observe that $0 \leq X_0, X_1 \leq np$, where $p = 2^{-c}$, and that U and V have exactly $2np - X_0 - X_1$ separating positions. Thus, the two coalitions are nonseparated when

both X_0 and X_1 equal np. Let us denote p_c the probability of this event. Then,

$$\begin{aligned} p_c &= \Pr\{X_0 = np, X_1 = np\} = \Pr\{X_0 = np\} \Pr\{X_1 = np | X_0 = np\} \\ &= \Pr\{X_0 = np\} \sum_{j=0}^{np} \Pr\{Y_0 = j | X_0 = np\} \Pr\{X_1 = np | X_0 = np, Y_0 = j\}. \end{aligned} \quad (4.7)$$

Here, the auxiliary r.v. Y_0 counts the number of nonseparating positions i such that $P_i(U) = P_i(V) = \{0\}$, i.e., $u_i^1 = \cdots = u_i^c = v_i^1 = \cdots = v_i^c = 0$.

Now, let us denote $h(k; N, K, n)$ the pmf at k of a hypergeometric r.v. with a total size of the population N, number of items with the desired characteristic K, and number of samples drawn n. Then,

$$h(k; N, K, n) \stackrel{\text{def}}{=} \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}, \qquad 0 \le k \le n.$$

It is not difficult to see that all the probabilities that appear in (4.7) can be expressed in terms of the hypergeometric pmf as follows:

$$\begin{aligned} \Pr\{X_0 = np\} &= h(np; n, n - np, np), \\ \Pr\{Y_0 = j | X_0 = np\} &= h(j; n - np, np, np), \\ \Pr\{X_1 = np | X_0 = np, Y_0 = j\} &= h(np; n - np, n - 2np + j, np). \end{aligned}$$

Expanding the terms of the hypergeometric pmf, equation (4.7) reduces to

$$p_c = \frac{1}{\binom{n}{np}\binom{n-np}{np}} \sum_{j=0}^{np} \binom{n - 2np + j}{np} \binom{np}{j} \binom{n - 2np}{np - j}.$$

Considering the generalization of the binomial coefficient to real values, and using its well-known asymptotic form

$$\lim_{n\to\infty} \frac{1}{n} \log_2 \binom{n}{p\,n} = H_2(p), \qquad 0 \le p \le 1, \quad (4.8)$$

we obtain for n increasing

$$\lim_{n\to\infty}\frac{1}{n}\log_2 p_c \le \max_{0\le j'\le p}\Big\{(1-2p+j')H_2\big(\tfrac{p}{1-2p+j'}\big)+p\,H_2\big(\tfrac{j'}{p}\big)+ (1-2p)H_2\big(\tfrac{p-j'}{1-2p}\big)-H_2(p)-(1-p)H_2\big(\tfrac{p}{1-p}\big)\Big\}.$$

It is routine to check that the j' that maximizes the expression is $j'=\ell$, and hence the lemma follows. □

Theorem 4.9. For the maximal possible asymptotical rate $R_2^{\mathrm{sep}^*}(c)$ among all asymptotically almost (c,c)-separating families of binary codes we have

$$R_2^{\mathrm{sep}^*}(c) \ge -\frac{1}{c}\,G(c).$$

Proof. Consider a random binary (n,M)-code C. Note that, according to (4.6), the expected ratio E of typical sets has $\lim_{n\to\infty} E = 1$. Hence, it can be considered that all c-subsets are δ-typical in the limit.

Let $U,V\subseteq C$ be two δ-typical c-subsets. Moreover, let p'_c be the probability that U and V are nonseparated. Hence the expected number of nonseparated "couples of c-subsets" $\{U,V\}$, where U and V are δ-typical, is at most $\binom{M}{c}\binom{M-c}{c}p'_c$, and the probability ε that a given δ-typical c-subset U is not separated satisfies

$$\varepsilon \le \binom{M-c}{c}p'_c.$$

Hence, using Lemma 4.8,

$$\lim_{n\to\infty}\frac{1}{n}\log_2\varepsilon \le c\,R + G(c).$$

Take a sequence of codes $\mathcal{C}=(C_i)_{i\ge1}$ of growing length such that each (n_i,M_i)-code C_i is a random code. Again, the probabilistic argument above shows that taking $\delta_i=\delta_i(n_i)=\ln n_i/\sqrt{n_i}$ there exists a sequence of codes with $\lim_{i\to\infty}\varepsilon_i=0$ for any rate

$$R < -\frac{1}{c}\,G(c).$$ □

4.3 Almost Secure Frameproof Codes

In this section we relax the definition of separating (or secure frameproof) code, again, in order to obtain better code rates. The notion that we introduce here allows us to separate the concepts of almost separating and almost secure frameproof codes.

Let us call a vector $\mathbf{z} \in \mathrm{desc}_c(C)$ *c-uniquely decodable* if $\mathbf{z} \in \mathrm{desc}(U)$ for some c-subset $U \subseteq C$ and $\mathbf{z} \notin \mathrm{desc}(V)$ for any c-subset $V \subseteq C$ such that $U \cap V = \emptyset$. Note that the c-secure frameproof codes from Definition 4.2 can be regarded as codes where all vectors $\mathbf{z} \in \mathrm{desc}_c(C)$ are c-uniquely decodable. This alternate definition allows us to introduce the following concept.

Definition 4.10. A code $C \subseteq Q^n$ is *ε-almost c-secure frameproof* if the ratio of c-uniquely decodable vectors among all $\mathbf{z} \in \mathrm{desc}_c(C)$ is at least $1 - \varepsilon$.

A sequence of codes $\mathcal{C} = (C_i)_{i \geq 1}$ of growing length n_i is an *asymptotically almost c-secure frameproof family* if every code C_i is an ε_i-almost c-secure frameproof code and $\lim_{i \to \infty} \varepsilon_i = 0$.

Consider again the asymptotical rate of a sequence of codes (4.3). As above, we are interested in estimating the maximal possible asymptotical rate, $R_q^{\mathrm{SFP}^*}(c)$, among all asymptotically almost c-secure frameproof families.

Theorem 4.11. For the maximal possible asymptotical rate $R_q^{\mathrm{SFP}^*}(c)$ among all asymptotically almost c-secure frameproof families of codes we have

$$R_q^{\mathrm{SFP}^*}(c) \geq -\frac{1}{c} \log_q(1 - (1 - 1/q)^c).$$

Proof. Consider a random (n, M)-code C over a q-ary alphabet. Also, consider a vector $\mathbf{z} = (z_1, \ldots, z_n)$ which is generated by a c-coalition $U \subseteq C$. For a random c-coalition $V \subseteq C$ such that $U \cap V = \emptyset$, using Lemma 4.4, we have

$$\Pr\{\mathbf{z} \in \mathrm{desc}(V)\} = (1 - p_{q,c,1}^{\mathrm{disj.}})^n = (1 - (1 - 1/q)^c)^n.$$

In fact, there are n positions, and the probability that each position $1 \leq i \leq n$ satisfies $y_i \notin P_i(V)$ equals $p_{q,c,1}^{\mathrm{disj.}}$, because V is a random, independent coalition from

U. Therefore the probability that a given vector $\mathbf{z} \in \mathrm{desc}_c(C)$ is not c-uniquely decodable is at most $\varepsilon \leq M^c(1 - p^{\mathrm{disj.}}_{q,c,1})^n$. Hence, there is a sequence $\mathcal{C} = (C_i)_{i \geq 1}$ of growing length n_i such that for each (n_i, M_i)-code C_i the ratio of c-uniquely decodable vectors among all $\mathbf{z} \in \mathrm{desc}_c(C_i)$ is at least $1 - \varepsilon_i \geq 1 - M_i^c(1 - p^{\mathrm{disj.}}_{q,c,1})^{n_i}$. Taking $M_i = o((1 - p^{\mathrm{disj.}}_{q,c,1})^{-n_i/c})$, i.e., $M_i = o((1 - (1 - 1/q)^c)^{-n_i/c})$, we have $\lim_{i \to \infty} \varepsilon_i = 0$, and the proof follows. □

Remark 4.12. If $C \subseteq Q^n$ is an ε-almost c-secure frameproof code, then for the family of codes $\varphi(C)$, where φ runs over the group G of all isometries of the Hamming space Q^n, the probability that any given vector $\mathbf{y}$ can be generated by two disjoint coalitions is at most ε (since the group G is twice transitive). This property allows us to replace the (c, c)-separating codes in the main construction of fingerprinting codes from [15] with asymptotically almost c-secure frameproof families, what will result in larger code rate with the same polynomial complexity identification algorithm. See Section 4.5 below.

Remark 4.13. For the case of a family of codes (instead of a single code) we can say "probability" instead of "ratio." Namely, for *every* "received" vector $\mathbf{y}$ the probability (i.e., the "ratio" of codes) that there exist at least two different c-coalitions U, V of codewords which can generate $\mathbf{y}$, is at most ε. Then, of course, for $c = 2$ the lower bound on the code rate is the same and it also follows from [55].

4.3.1 Geometric Interpretation

For an (n, M)-code C over Q, consider the set of convex combinations between two vectors $\mathbf{u}, \mathbf{v} \subseteq C$ as

$$\{\mathbf{z} \in Q^n : d(\mathbf{u}, \mathbf{z}) + d(\mathbf{z}, \mathbf{v}) = d(\mathbf{u}, \mathbf{v})\}. \tag{4.9}$$

Note that for a c-subset $U \subseteq C$, its *convex hull* $[U] \subseteq Q^n$, i.e., the smallest set containing all convex combinations between any two of its elements, is precisely the envelope under the narrow-sense model, $\mathrm{desc}(U)$. Therefore, for the case $c = 2$ and $U = \{\mathbf{u}, \mathbf{v}\} \subseteq C$, equation (4.9) suggests calling the set $[\{\mathbf{u}, \mathbf{v}\}]$ a *segment* of C with

vertices $\mathbf{u}$ and $\mathbf{v}$. For $c = 3$ and a 3-coalition $U \subseteq C$, the set $[U]$ could be called a (convex) *polygon*, and so on. For arbitrary c, let us call $[U]$ a (convex) *c-polytope*.

Hence, a c-secure frameproof code, or, what is the same, a (c, c)-separating code, can be regarded as a set of points C in the q-ary Hamming space Q^n with the property that any two c-polytopes $[U]$, $[V]$ with $U, V \subseteq C$ do not intersect, provided that they do not share a common vertex from C.

For a random binary code C, consider the union $C^{[c]}$ of all points generated from c-polytopes $[U]$ such that $U \subseteq C$, as in the proof of Theorem 4.11. In other words, $C^{[c]} = \mathrm{desc}_c(C)$. For a given $\mathbf{z} \in Q^n$ and a random c-subset $V \subseteq C$, let us call

$$g(n) = \Pr\{\mathbf{z} \in [V]\} = \Pr\{\mathbf{z} \in \mathrm{desc}(V)\} = (1 - p_{q,c,1}^{\mathrm{disj.}})^n,$$

which follows from the proof of Theorem 4.11 above. Hence, the size of $C^{[c]}$ can be estimated as

$$|C^{[c]}| = \sum_{\mathbf{z} \in Q^n} \Pr\{\mathbf{z} \in C^{[c]}\} = q^n \Pr\{\mathbf{z} \in C^{[c]}\} = q^n (1 - (1 - g(n))^{\binom{M}{c}}). \tag{4.10}$$

Now, let us define the "volume" of $C^{[c]}$ by counting every point $\mathbf{z} \in C^{[c]}$ with its multiplicity, i.e., the number of c-polytopes to which $\mathbf{z}$ belongs. Using (4.2), we have

$$|C^{[c]}| \le \mathrm{vol}(C^{[c]}) = \binom{M}{c} \Big(\sum_j j\, v(j) \Big)^n = \binom{M}{c} q^n\, g(n). \tag{4.11}$$

This result can be obtained in two different but equivalent ways. Indeed, there are $\binom{M}{c}$ c-polytopes, and the probability that each $\mathbf{z} \in Q^n$ is generated by a given c-polytope $[U]$ is $g(n)$. Alternatively, the average number of points generated by every

c-polytope $[U]$ can be computed as

$$\begin{aligned}
&\sum_{j_1=1}^{c} \cdots \sum_{j_n=1}^{c} j_1 \cdots j_n \Pr\{|P_1(U)| = j_1, \ldots, |P_n(U)| = j_n\} \\
&\quad = \sum_{j_1} \cdots \sum_{j_n} j_1 \cdots j_n \, v(j_1) \cdots v(j_n) = \Big(\sum_j j \, v(j)\Big)^n = \Big(q^{-c} \sum_j j \, q^{\underline{j}} \left\{ {c \atop j} \right\}\Big)^n \\
&\quad \overset{\text{(a)}}{=} \Big(q^{-c} \sum_j q(q^{\underline{j}} - (q-1)^{\underline{j}}) \left\{ {c \atop j} \right\}\Big)^n \overset{\text{(b)}}{=} (q^{-c+1}(q^c - (q-1)^c))^n = q^n \, g(n).
\end{aligned}$$

Here, (a) is obtained by routine algebraic manipulation, and (b) follows from the well-known identity $x^c = \sum_j x^{\underline{j}} \left\{ {c \atop j} \right\}$.

Hence, from (4.10) and (4.11) two nontrivial observations can be drawn. First, for $M = o(g(n)^{-1/c})$, we have $\lim_{n\to\infty} \mathrm{vol}(C^{[c]})/|Q^n| = 0$, i.e., the volume of $C^{[c]}$ is relatively small compared to the volume of the whole Hamming space. Second, consider the average asymptotical multiplicity of the points from $C^{[c]}$,

$$\lim_{n\to\infty} \frac{\mathrm{vol}(C^{[c]})}{|C^{[c]}|} = \lim_{n\to\infty} \frac{M^c \, g(n)}{1 - (1 - g(n))^{M^c}} = \lim_{n\to\infty} \frac{M^c \, g(n)}{1 - e^{-M^c \, g(n)}}.$$

The last equality follows from the fact that $\lim_{n\to\infty} g(n) = 0$. Taking again $M = o(g(n)^{-1/c})$, it is easy to see that the the main part of points from $C^{[c]}$ have multiplicity 1, i.e., covered only once by code polytopes, which is a stronger statement than Theorem 4.11.

4.4 Comparison of Results

In Table 4.1 we give some figures for the lower bounds on the asymptotical rate of q-ary separating, almost separating and almost secure frameproof codes. It can be seen that the lower bounds on the rate for almost separating codes roughly doubles the rate of ordinary separating codes. This proportion increases for c growing and slightly decreases for q growing, staying at about 1.9 for $c > 7$, almost independent of q.

q	Code	$c = 2$	3	4	5	10	15
	Separating	6.422E−2	9.161E−3	1.616E−3	3.134E−4	1.448E−7	9.266E−11
2	Almost sep.	1.038E−1	1.605E−2	2.910E−3	5.725E−4	2.753E−7	1.792E−10
	Almost sep.(*)	1.422E−1	1.703E−2	3.001E−3	5.815E−4	2.754E−7	1.792E−10
	Almost s.f.	2.075E−1	6.422E−2	2.328E−2	9.161E−3	1.410E−4	2.935E−6
	Separating	7.625E−2	1.080E−2	1.796E−3	3.191E−4	8.433E−8	2.997E−11
3	Almost sep.	1.249E−1	1.948E−2	3.320E−3	5.954E−4	1.609E−7	5.798E−11
	Almost s.f.	2.675E−1	1.066E−1	5.008E−2	2.571E−2	1.592E−3	1.387E−4
	Separating	9.562E−2	1.561E−2	2.889E−3	5.624E−4	2.327E−7	1.415E−10
4	Almost sep.	1.524E−1	2.772E−2	5.288E−3	1.040E−3	4.428E−7	2.735E−10
	Almost s.f.	2.982E−1	1.318E−1	6.860E−2	3.908E−2	4.181E−3	6.470E−4
	Separating	1.114E−1	2.091E−2	4.307E−3	9.053E−4	4.067E−7	2.158E−10
5	Almost sep.	1.744E−1	3.671E−2	7.853E−3	1.674E−3	7.741E−7	4.173E−10
	Almost s.f.	3.174E−1	1.486E−1	8.185E−2	4.934E−2	7.058E−3	1.484E−3
	Separating	1.549E−1	4.329E−2	1.350E−2	4.201E−3	6.568E−6	5.615E−9
10	Almost sep.	2.357E−1	7.372E−2	2.419E−2	7.728E−3	1.251E−5	1.086E−8
	Almost s.f.	3.606E−1	1.890E−1	1.159E−1	7.755E−2	1.862E−2	6.675E−3
	Separating	1.752E−1	5.723E−2	2.162E−2	8.418E−3	4.303E−5	7.895E−8
15	Almost sep.	2.649E−1	9.653E−2	3.840E−2	1.539E−2	8.202E−5	1.527E−7
	Almost s.f.	3.783E−1	2.064E−1	1.313E−1	9.098E−2	2.572E−2	1.081E−2

Table 4.1: Lower bounds on the rate of some q-ary codes. The lower bounds (*) correspond to the analysis for the binary case from Section 4.2.3.

Also, the refined analysis for binary almost separating codes of Section 4.2.3 shows an improvement on the lower bound, especially for small values of c.

4.5 Application to Fingerprinting Codes

In this section, we show how binary almost separating or almost secure frameproof codes can be used to construct a family of binary fingerprinting codes. We will outline the code construction and derive existence conditions. Our work has been built upon [15] to obtain codes with better rates.

4.5.1 Family Construction

Recall that for a fingerprinting scheme to achieve an error probability as small as desired a single code is not sufficient, but a family of codes $\mathcal{C} = \{C_t\}_{t\in T}$ is needed. As in Section 3.4, we will proceed by modifying the codes obtained in Construction 2.13 to obtain a family of binary concatenated fingerprinting codes $\mathcal{C} = \{C_t\}_{t\in T}$ with error probability decreasing exponentially with the code length.

Also, as opposed to Construction 3.14, we will only use a single inner binary (l, q)-code C_{in}. This time, the randomness comes from the particular choice of the mappings from the inner code to Q, the outer code alphabet, $\phi : C_{\text{in}} \to Q$. Consider the vector of mappings $(\phi_1, \ldots, \phi_n)$, where $\phi_i, 1 \leq i \leq n$ are bijections between C_{in} and Q. It is clear that there are $(q!)^n$ different such vector mappings. If the mappings are arbitrarily numbered from 1 to $(q!)^n$, then

$$\Phi_t \stackrel{\text{def}}{=} (\phi_{t1}, \ldots, \phi_{tn}) \tag{4.12}$$

denotes the mapping indexed by t.

Construction 4.14. Let C_{out} be an (n, M)-code over a q-ary alphabet Q, and let C_{in} be a binary (l, q)-code. Also, let Φ_t denote the mapping indexed by t as in (4.12). Denote by C_t the code constructed in the following way:

$$C_t \stackrel{\text{def}}{=} \{\Phi_t(\mathbf{w}) : \mathbf{w} \in C_{\text{out}}\},$$

where

$$\Phi_t(\mathbf{w}) \stackrel{\text{def}}{=} (\phi_{t1}(w_1), \ldots, \phi_{tn}(w_n)).$$

The set $\mathcal{C} = \{C_t\}_{t \in T}$, with $T = \{1, \ldots, (q!)^n\}$, constitutes the concatenated family.

Again, to use the family from Construction 4.14, $\mathcal{C} = \{C_t\}_{t \in T}$, the distributor has to choose a secret value, $t \in T$ according to a pmf π. Each user is then assigned a codeword from C_t.

Let $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subseteq C_t$ denote a c-coalition, and let $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^c\} \subseteq C_{\text{out}}$ be the subset of their corresponding outer codewords. That is, $\mathbf{u}^j = \Phi_t(\mathbf{w}^j)$ for $1 \leq j \leq c$. Also, let

$$\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n}) \in \text{desc}(U),$$

be a descendant created by coalition U.

In the discussion of the identification algorithm, we will consider that the identification process of each inner block $\mathbf{z}_i$ returns a set $V_i \subseteq C_{\text{in}}$ of at most c codewords, such that $\mathbf{z}_i \in \text{desc}(V_i)$. Observe that, if the inner code C_{in} is an ε-almost (c, c)-separating or an ε-almost c-secure frameproof code, then with probability $\geq 1 - \varepsilon$ there is a $\mathbf{v} \in V_i$ such that $\mathbf{v}$ agrees with the ith block of a traitor codeword, i.e., $\mathbf{v} = \phi_{ti}(w_i)$ for some $\mathbf{w} = (w_1, \ldots, w_n) \in W$.

We now state, in the form of a theorem, the precise parameters of the codes in Construction 4.14 so that we can achieve exponentially small error probability when used in conjunction with Algorithm 4.1.

Theorem 4.15. Let C_{out} be an (n, M)-code over a q-ary alphabet Q with minimum distance $d = d(C_{\text{out}})$, and let C_{in} be an ε_{in}-almost (c, c)-separating or an ε_{in}-almost c-secure frameproof (l, q)-code. Let $\mathcal{C} = \{C_t\}_{t \in T}$ be the family of concatenated codes from Construction 4.14 with outer code C_{out}, inner code C_{in}, the mappings Φ_t, the set of keys T, and $\pi(t) = |T|^{-1}$. For $q > c^2$, if

$$d > n - \frac{n(1-\sigma)}{c^2} + \frac{n(c-1)}{c(q-c)}, \qquad \text{with } \varepsilon_{\text{in}} < \sigma < \frac{q - c^2}{q - c}, \tag{4.13}$$

Algorithm 4.1 Concatenated Tracing Algorithm 3

Input: A concatenated code C_t from Construction 4.14, satisfying the conditions from Theorem 4.15, and a descendant $\mathbf{z} \in \mathrm{desc}_c(C_t)$,

$$\mathbf{z} = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n}).$$

Output: A subset of codewords of C_t.

1) For $1 \leq i \leq n$, decode each block $\mathbf{z}_i = (z_{i1}, \ldots, z_{il})$ of the the descendant $\mathbf{z}$ as follows:

   (a) Find all c-subsets $V \subseteq C_{\mathrm{in}}$ such that $\mathbf{z}_i \in \mathrm{desc}(V)$.

   (b) If the intersection of all c-subsets V found in Step 1a) is empty, set $\mathcal{Z}_i = \emptyset$.

   (c) Otherwise, pick an arbitrary c-subset V from Step 1a) and use the inverse mapping
   $$\phi_{ti}^{-1} : C_{\mathrm{in}} \to Q$$
   to obtain a set $\mathcal{Z}_i$ of c symbols from Q.

2) Construct the set vector
   $$\boldsymbol{\mathcal{Z}} := (\mathcal{Z}_1, \ldots, \mathcal{Z}_n).$$

3) For each $\mathbf{w} \in C_{\mathrm{out}}$, compute the similitude $\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}})$, according to (2.1).

4) Output the set $L := \{\mathbf{u}^1, \ldots, \mathbf{u}^s\}$, consisting of all codewords $\mathbf{u} = \Phi_t(\mathbf{w}) \in C_t$, such that
   $$\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c},$$
   for some codeword $\mathbf{w} \in C_{\mathrm{out}}$. If $L = \emptyset$, declare identification error.

then the family of concatenated codes $\mathcal{C} = \{C_t\}_{t\in T}$ together with Algorithm 4.1 is a c-secure with ε-error family of binary codes, with exponentially small error,

$$\varepsilon \leq q^k 2^{-nD(\rho\|\frac{c-1}{q-c})} + 2^{-nD(\sigma\|\varepsilon_{\text{in}})} = \exp(-\Omega(n)), \tag{4.14}$$

where $\rho = \frac{1-\sigma}{c} - c(1-d/n)$.

Proof. Let $U \subseteq C_t$ be a c-coalition, and let $W \subseteq C_{\text{out}}$ be the subset of their corresponding outer codewords, as stated above. Also, let $\mathbf{z}$ be

$$\mathbf{z} = (\underbrace{z_{11}, \ldots, z_{1l}}_{\mathbf{z}_1}, \ \ldots, \ \underbrace{z_{n1}, \ldots, z_{nl}}_{\mathbf{z}_n})$$

a descendant created by coalition U.

First, note that in Step 1b) of Tracing Algorithm 1 we are discarding all "nonseparating blocks" by setting $\mathcal{Z}_i = \emptyset$, an event that occurs with probability $\leq \varepsilon$ due to the properties of the inner code. Hence, $\mathcal{Z}_i \cap P_i(W) \neq \emptyset$, i.e., $\mathcal{Z}_i$ contains at least one element w_i for some $\mathbf{w} = (w_1, \ldots, w_n) \in W$, with probability $\geq 1 - \varepsilon$.

Let X be the number of discarded blocks, which can be upper bounded using a binomial r.v. of parameters n and $p \leq \varepsilon_{\text{in}}$. Since $\sigma > \varepsilon_{\text{in}}$, we can use (2.10) to see that

$$\Pr\{X \geq n\sigma\} \leq 2^{-nD(\sigma\|\varepsilon_{\text{in}})}, \tag{4.15}$$

which decreases exponentially with n.

That is, with high probability, there is some coalition codeword $\hat{\mathbf{u}} = \Phi_t(\hat{\mathbf{w}}) \in U$ for some $\hat{\mathbf{w}} \in W$ such that

$$\mathrm{s}(\hat{\mathbf{w}}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c}, \tag{4.16}$$

hence, a traitor is identified.

On the other hand, for an innocent codeword $\mathbf{u} = \Phi_t(\mathbf{w})$, i.e., $\mathbf{w} \notin W$, if the element w_i appears in a nondiscarded set $\mathcal{Z}_i$ it could be because $w_i \in P_i(W)$. Since any two codewords of C_{out} can agree in $\leq n - d$ positions, this event can happen in at most $c(n-d)$ positions. Also, whenever $w_i \notin P_i(W)$ the probability that $w_i \in \mathcal{Z}_i$

can be bounded as

$$p_i = \Pr\{w_i \in \mathcal{Z}_i | w_i \notin P_i(W)\} \leq \frac{c-1}{q-c}. \tag{4.17}$$

For $1 \leq i \leq n$, let Y_i be an r.v. that takes the value 1 with probability p_i and 0 with probability $1 - p_i$. Therefore for $\mathbf{w} \notin W$,

$$\begin{aligned}
\Pr\Big\{\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c}\Big|\mathbf{w} \notin W\Big\} &\leq \Pr\Big\{c(n-d) + \sum_{i=1}^{\substack{n-X-\\ c(n-d)}} Y_i \geq n\frac{1-\sigma}{c}\Big\} \\
&\leq \Pr\Big\{c(n-d) + \sum_{i=1}^{n} Y_i \geq n\frac{1-\sigma}{c}\Big\} = \Pr\Big\{\sum_{i=1}^{n} Y_i \geq n\rho\Big\} \\
&\overset{\text{(a)}}{\leq} \Pr\Big\{Y \geq n\rho\Big\} \leq 2^{-nD(\rho\|\frac{c-1}{q-c})}.
\end{aligned}$$

Inequality (a) above follows from (4.17), by comparing the summation $\sum_{i=1}^{n} Y_i$ with an appropriate binomial r.v. Y of parameters n and $(c-1)/(q-c)$. Also, since $(c-1)/(q-c) < \rho$, which is implied by the condition in the minimum distance of the outer code (4.13), applying (2.10) again gives the last inequality above.

Since there are q^k codewords, the probability of accusing an innocent user as guilty is upper bounded as

$$\begin{aligned}
&\Pr\Big\{\max_{\mathbf{w}\notin W} \mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c}\Big\} \\
&\quad\leq q^k \Pr\Big\{\mathrm{s}(\mathbf{w}, \boldsymbol{\mathcal{Z}}) \geq n\frac{1-\sigma}{c}\Big|\mathbf{w} \notin W\Big\} \\
&\quad\leq q^k 2^{-nD(\rho\|\frac{c-1}{q-c})}.
\end{aligned} \tag{4.18}$$

Recall that the probability of not accusing a real traitor is (4.15). Putting this together with (4.18), we have

$$\varepsilon \leq q^k 2^{-nD(\rho\|\frac{c-1}{q-c})} + 2^{-nD(\sigma\|\varepsilon_{\mathrm{in}})}.$$

Moreover, this shows that with error probability ε no codeword $\mathbf{w} \notin W$ will lie within the decoding radius (4.16). □

4.5.2 Existence Conditions

The existence of a family of fingerprinting codes with error probability decreasing exponentially in the outer code length is guaranteed using similar arguments to those from [15]. Using Reed-Solomon as outer codes we have the following result, which assumes c fixed and q growing.

Corollary 4.16. Let C_{out} be an extended $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$ of rate $R_{\text{out}} = R(C_{\text{out}})$, and let C_{in} be a binary ε_{in}-almost (c, c)-separating or ε_{in}-almost c-secure frameproof (l, q)-code of rate $R_{\text{in}} = R(C_{\text{in}})$. Let $\mathcal{C} = \{C_t\}_{t \in T}$ be the family of concatenated codes from Construction 4.14 with outer code C_{out}, inner code C_{in}, the mappings Φ_t, the set of keys T, and $\pi(t) = |T|^{-1}$. For any $q > c^2$, and any

$$R_{\text{out}} < \frac{1-\sigma}{c(c+1)}, \qquad \text{with } \varepsilon_{\text{in}} < \sigma < \frac{q-c^2}{q-c}, \tag{4.19}$$

the family of concatenated codes $\mathcal{C} = \{C_t\}_{t \in T}$ together with Algorithm 4.1 is a c-secure with ε-error family of binary codes, of rate

$$R = R_{\text{out}}\, R_{\text{in}},$$

and error probability ε decreasing exponentially as

$$\varepsilon \leq 2^{-n\, l(\frac{1-\sigma}{c} R_{\text{in}} - (c+1)R + o(1))} + 2^{-nD(\sigma \| \varepsilon_{\text{in}})}.$$

Proof. If C_{out} is an extended Reed-Solomon code with minimum distance d, we have $n = q$ and $1 - d/n = R_{\text{out}} - 1/n$, hence from Theorem 4.15

$$\rho = \frac{1-\sigma}{c} - c\Big(R_{\text{out}} - \frac{1}{n}\Big). \tag{4.20}$$

Now, the error probability from (4.14) can be expressed as

$$\varepsilon \leq 2^{-n\,l\,R_{\text{in}}((\log_2 q)^{-1}D(\rho\|\frac{c-1}{q-c})-R_{\text{out}})} + 2^{-nD(\sigma\|\varepsilon_{\text{in}})}.$$

The proof follows after substituting (4.20) into the previous equation and taking into account that

$$\lim_{q\to\infty}(\log_2 q)^{-1}D\Big(\rho\Big\|\frac{c-1}{q-c}\Big) = \rho$$

for c fixed and q growing. □

Besides Reed-Solomon codes, in [15] algebraic-geometric codes are also proposed as outer codes. As noted in Section 4.4, replacing ordinary separating codes by almost separating codes enables us to double the asymptotical rate of the fingerprinting codes proposed in [15].

4.5.3 Efficient Decoding

Finally, it is worth noting here that the main reason for Construction 4.14, Theorem 4.15 and Corollary 4.16 is to mimic the following strategy from [15]. If the outer code C_{out} is a Reed-Solomon (or an algebraic-geometric code), then traitor identification can be efficiently done in polynomial time by using the list-decoding algorithms from [41].

We now show how using the Kötter-Vardy algorithm, as in Section 3.4.1, all codewords $\mathbf{w} \in C_{\text{out}}$ that satisfy (4.16) can be found in polynomial time. To see this, for $1 \leq j \leq q$ and $1 \leq i \leq n$, set up a q-by-n reliability matrix $\mathcal{R} = (r_{ji})$ as

$$r_{ji} = \begin{cases} \dfrac{1-\xi}{c} & \text{if } \alpha_j \in \mathcal{Z}_i, \\ \dfrac{\xi}{q-c} & \text{if } \alpha_j \notin \mathcal{Z}_i \text{ and } \mathcal{Z}_i \neq \emptyset, \\ \dfrac{1}{q} & \text{otherwise.} \end{cases}$$

If there are no more than $n\sigma$ empty sets $\mathcal{Z}_i$, the condition of successful decoding for $\mathbf{w}$ reduces to

$$\frac{(n-n\sigma)\frac{1-\xi}{c}+n\sigma\frac{1}{q}}{\sqrt{(n-n\sigma)\frac{(1-\xi)^2}{c}+n\sigma\frac{1}{q}}} > \sqrt{\frac{n(1-\sigma)}{c^2}+\frac{n(c-1)}{c(q-c)}}. \tag{4.21}$$

It is easy to see that the left-hand side in the condition above is maximized for $\xi = 0$. This matches the intuition. By the almost (c, c)-separating or almost c-secure frameproof property of the inner code, we conclude that in any nonempty subset $\mathcal{Z}_i$, at least one of the symbols matches an element from $P_i(W)$. Setting $\xi = 0$ means that the error needs only to be "spread" among the elements of $\mathcal{Z}_i$. Under this circumstance, condition (4.21) is met when the outer code satisfies (4.19).

4.6 Conclusion

In this chapter we have presented two different relaxed versions of (c, c)-separating codes, namely almost (c, c)-separating and almost c-secure frameproof codes. The notions introduced allows us to separate two concepts that coincide in the case of absolute separation.

To show existence bounds for almost (c, c)-separating codes we have used the concept of typicality. Two distinct approaches are considered. In the first approach, we consider that a typical set of at most c codewords is separated with very high probability, with all other disjoint sets also of at most c codewords. This analysis shows that there exists almost (c, c)-separating codes that double the asymptotical rate of ordinary separating codes. In the second approach, we have used a refined analysis, applicable to the binary case, which allows us to show the existence of codes with even better rates.

For almost c-secure frameproof codes we have used a probabilistic analysis showing that there exist c-secure frameproof codes with asymptotical rate four times the asymptotical rate of ordinary (c, c)-separating codes.

We believe that these two notions are essentially different, in particular, we conjecture that for asymptotical rates

$$R_q^{\mathrm{SFP}^*}(c) > R_q^{\mathrm{sep}^*}(c),$$

but it could be a rather difficult question since even for the simplest case $q = c = 2$ the best upper bound for the rate of $(2,2)$-separating codes $\overline{R}_2^{\mathrm{sep}}(2,2) \leq 0.2835$ is very far from being "useful."

Finally, we have presented a concatenated construction of a family of fingerprinting codes. The use of almost (c,c)-separating codes as inner codes allows us to obtain better rates preserving the exponential decline of the error probability on the outer code length, and it also allows us to obtain a polynomial-time identification algorithm.

The results of this chapter have been published in [4, 5, 9].

Chapter 5

Construction of Almost Secure Frameproof Codes

In this chapter we discuss the construction of almost secure frameproof codes over binary alphabets. Recall that the notions of separating and secure frameproof code coincide when we are considering their ordinary version. Relaxing the definition of a separating code in two different ways allows us to obtain two different notions, as it was shown in Chapter 4, where we showed their application to fingerprinting schemes. For instance, they are useful to construct a family of fingerprinting codes in the style of [15], improving the lower bound on the asymptotical rate.

We will connect the concept of almost secure frameproof code from Definition 4.10 with the concept of weakly biased arrays [56], which is closely related to small-bias probability spaces [57, 58]. Let us consider an n-by-M matrix $A = (a_{ij})$ with entries from $\mathbb{F}_2$, which is commonly called a *binary array*. Also, for each subset of indices $S \subseteq \{1, \ldots, M\}$, let us call the sum $\sum_{j \in S} a_{ij}$ the *parity vector* of S. The array A is *weakly biased* if the parity vector of every subset S has, approximately, the same number of zeros and ones. Note that if A contains every possible row from $\mathbb{F}_2^M$ repeated the same number of times, then A can be regarded as unbiased.

If A is weakly biased and $(X_1, \ldots, X_M)$ is a random vector generated by choosing a row of a binary array A uniformly at random, then the r.v.'s $X_1, \ldots, X_M$ are readily seen to be "almost" independent. Let $S = \{i_1, \ldots, i_s\} \subseteq \{1, \ldots, M\}$ be a subset of

$s \leq t$ indices. We say that the array A is *ε-away from t-wise independence* if the induced probability distribution on the r.v.'s $X_{i_1}, \ldots, X_{i_s}$ is "close" to the uniform distribution on $\mathbb{F}_2^s$ for every possible subset S of size at most t.

Since every subset of t columns of an ε-away from t-wise independence array A generates an "almost" uniform distribution on $\mathbb{F}_2^t$, then the array A has an interesting property. For a small enough value of ε, every $\mathbb{F}_2^t$-configuration, i.e., every vector from $\mathbb{F}_2^t$, appears in every subset of t columns. A set of rows (vectors) satisfying this property constitute what is known as an *(M, t)-universal set*. This observation will prove very useful for our purposes, since for $t = 2c$ an (M, t)-universal set of size n immediately generates a (c, c)-separating code.

From Definitions 4.6 and 4.10 it is easy to see that a code is a (c, c)-separating code if and only if it is a c-secure frameproof code. However, when the definitions of separation and frameproofness are relaxed, then both notions are different. Intuitively it seems clear that almost separation is a more strict requirement than almost secure frameproofness. In fact, we already showed in Chapter 4 that there exist almost secure frameproof codes with a much higher rate than almost separating codes [4]. The strategy used to establish the existing lower bounds in the asymptotical rates of almost separating and almost secure frameproof codes relies on a standard probabilistic argument. It has been shown that there exist codes that achieve such rates within an ensemble of codes, in which every codeword $\mathbf{u} = (u_1, \ldots, u_n)$ has been chosen at random with $\Pr\{u_i = 0\} = \Pr\{u_i = 1\} = 1/2$ for each position $1 \leq i \leq n$.

We are now in the position to underline the structure of the chapter. In Section 5.1 we provide some useful definitions and a brief overview of previous results. The main contribution is discussed in Section 5.2. We begin by proving that the above choice of probabilities $\Pr\{u_i = 0\} = \Pr\{u_i = 1\} = 1/2$ is in fact the appropriate one to use to obtain codes with good separation properties. With this in mind we move into weakly biased arrays, where by adjusting the bias we provide explicit constructions for sets of vectors that are *almost* (M, t)-universal. Finally, we show that these constructions are useful to construct almost c-secure frameproof codes, which, using the results from Chapter 4, yield an explicit construction of a c-secure fingerprinting codes with small error and efficient identification algorithm.

5.1 Weakly Biased and Weakly Dependent Arrays

In this section we present the concepts about weakly biased and weakly dependent arrays that will be used in the constructions below. We will concentrate on the binary case, since our goal is to construct binary almost secure frameproof codes. Weakly biased and weakly dependent arrays are strongly related to small-bias probability spaces. For a more detailed exposition, we refer the reader to [56, 57, 58].

Consider the finite field $\mathbb{F}_2 = \{0, 1\}$. A *binary* (n, M)*-array* A is an n-by-M matrix whose entries are elements from $\mathbb{F}_2$. For a binary (n, M)-array A and a subset of indices $S \subseteq \{1, \ldots, M\}$ of size s, let us denote $\nu_S(\mathbf{a}; A)$ the number of rows of A whose projection onto the indices of S equals the vector $\mathbf{a} \in \mathbb{F}_2^s$. We will omit the subindex S whenever $s = M$, i.e., when we are considering the whole rows of the array. In particular, for a binary vector of length n, $\mathbf{u} \subseteq \mathbb{F}_2^n$, viewed as a binary $(n, 1)$-array, $\nu(0; \mathbf{u})$ and $\nu(1; \mathbf{u})$ denote its number of zeros and ones, respectively.

Definition 5.1. Let $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$. The *bias* of vector $\mathbf{u}$ is defined as

$$n^{-1}|\nu(0; \mathbf{u}) - \nu(1; \mathbf{u})|.$$

That is, a vector $\mathbf{u}$ which has approximately the same number of zeros and ones has small bias.

Definition 5.2. Let $0 \leq \varepsilon < 1$. A binary (n, M)-array is ε*-biased* if every nontrivial linear combination of its columns has bias $\leq \varepsilon$.

In other words, the bias of an array A is the bias of the linear binary code C generated by its columns. By definition, the bias of A is low if the bias of every nonzero codeword from C is low. Explicit constructions of ε-biased (n, M)-arrays, with $n = 2^{O(\log M + \log \frac{1}{\varepsilon})}$, can be found in [57].

The previous definition can be restricted by allowing a maximum number of columns in the linear combination.

Definition 5.3. Let $0 \leq \varepsilon < 1$. A binary (n, M)-array is t*-wise* ε*-biased* if every nontrivial linear combination of at most t columns has bias $\leq \varepsilon$.

We will also need the concepts of ε-dependent and ε-away from t-wise independence arrays.

Definition 5.4. Let $0 \leq \varepsilon < 1$. A binary (n, M)-array A is *t-wise ε-dependent* if for every subset $S \subseteq \{1, \ldots, M\}$ of $s \leq t$ columns and every vector $\mathbf{a} \in \mathbb{F}_2^s$, we have

$$|n^{-1}\nu_S(\mathbf{a}; A) - 2^{-s}| \leq \varepsilon.$$

Definition 5.5. Let $0 \leq \varepsilon < 1$. A binary (n, M)-array A is *ε-away from t-wise independence* if for every subset $S \subseteq \{1, \ldots, M\}$ of $s \leq t$ columns, we have

$$\sum_{\mathbf{a} \in \mathbb{F}_2^s} |n^{-1}\nu_S(\mathbf{a}; A) - 2^{-s}| \leq \varepsilon.$$

Remark 5.6. If the binary (n, M)-array A is t-wise ε-dependent, then it is $2^M\varepsilon$-away from t-wise independence, and if A is ε-away from t-wise independence, then it is t-wise ε-dependent.

As commented above, these definitions have an interpretation as a small-bias probability space [57,58]. If the r.v.'s $X_1, \ldots X_M$ take uniformly at random the corresponding values of a row of an (n, M)-array A that is ε-away from t-wise independence, then any t of the r.v.'s are "almost independent," provided that ε is small. Hence, one would like to obtain such arrays A with n (the size of the sample space) as small as possible.

For our purposes, the most important concept will be that of (M, t)-universal set. Now, we have the following definition.

Definition 5.7. An *(M, t)-universal set* B is a subset of $\mathbb{F}_2^M$ such that for every subset $S \subseteq \{1, \ldots, M\}$ of t positions the set of projections of the elements of B on the indices of S contains every $\mathbb{F}_2^t$-configuration.

Let A be a binary (n, M)-array. Observe that if for every subset $S \subseteq \{1, \ldots M\}$ of t columns and every vector $\mathbf{a} \in \mathbb{F}_2^t$ we have $\nu_S(\mathbf{a}; A) > 0$, then the rows of A form an (M, t)-universal set. We are interested in universal sets of as small size as possible.

In [57] the relationship between this concept and ε-away from t-wise independence arrays was shown.

Proposition 5.8. Let A be a binary (n, M)-array A. For $\varepsilon \leq 2^{-t}$, if A is ε-away from t-wise independence, then the rows of A yield an (M, t)-universal set of size n.

Moreover, the following result [57, 59, 60] also relates these concepts with the concept of ε-biased arrays.

Corollary 5.9. Let A be a binary (n, M)-array A. If A is ε-biased, then A is $2^{t/2}\varepsilon$-away from t-wise independence.

Hence, the construction of universal sets is reduced to the construction of ε-away from t-wise independence arrays by Proposition 5.8, which is reduced to the construction of ε-biased arrays by Corollary 5.9.

We will have occasion to use Corollary 5.9 in the next section, where an even more convenient method to construct ε-away from t-wise independence arrays will be discussed.

5.2 Constructions

In this section we present our constructions for almost secure frameproof codes. Before dwelling into explicit details we give an intuitive reasoning of our discussion.

First, we will show that the expected value of the probability that two c-coalitions are separated in a random binary code is maximized when the codewords are generated according to a probability vector $\mathbf{p} = (p_1, \ldots, p_n)$ such that $p_1 = \cdots = p_n = 1/2$. That is, we generate M random codewords $(u_1, \ldots, u_n)$ with $\Pr\{u_i = 1\} = p_i = 1/2$. But since we are interested in almost secure frameproof codes, we will be able to allow a small bias on these probabilities and therefore consider weakly biased arrays.

From [57], and by using the definitions and results from the previous section it can be seen that from weakly biased arrays we can obtain (M, t)-universal sets of size $n = \log_2 M \cdot 2^{O(t)}$. If we arrange the vectors of this universal set as the rows of an (n, M)-array, the columns of that matrix form a c-secure frameproof code for $t = 2c$.

This code has size M, length $n = \log_2 M \cdot 2^{O(2c)}$ and rate $2^{-O(2c)}$. The main idea is to allow a given number of $\mathbb{F}_2^t$-configurations in the universal set not to appear. This relaxation yields what we call an almost universal set. We finally prove that almost universal sets can be used to generate ε-almost c-secure frameproof codes with ε a function of the fraction of configurations allowed not to appear.

5.2.1 Separation in Random Codes

We start by making some observations about random codes. Let us assume that C is an (n, M)-random code generated according to a probability vector $\mathbf{p} = (p_1, \ldots, p_n)$, where $\mathbf{p}$ is chosen according to pmf $f_{\mathbf{p}}$. That is, we first generate a probability vector $\mathbf{p}$ of length n, distributed according to $f_{\mathbf{p}}$, and then we randomly generate M binary vectors $\mathbf{u} = (u_1, \ldots, u_n)$ such that $\Pr\{u_i = 1\} = p_i$. We would like to know which probability distribution $f_{\mathbf{p}}$ maximizes the probability that two c-coalitions are separated in a code generated in this way.

Lemma 5.10. Let C be an (n, M)-random code, whose codewords are generated according to the probability vector $\mathbf{p} = (p_1, \ldots, p_n)$. If the entries of $\mathbf{p}$ are iid r.v.'s, then the expected value of the probability that two c-coalitions are separated is maximized by taking $p_1 = \cdots = p_n = 1/2$.

Proof. Note that for a given $\mathbf{p}$ the probability that two c-coalitions U, V are not separated is $\prod_{i=1}^{n}(1 - 2p_i^c(1 - p_i)^c)$.

Hence, we have

$$E_{f_{\mathbf{p}}}\Big[1 - \prod_{i=1}^{n}(1 - 2p_i^c(1 - p_i)^c)\Big] = 1 - (1 - 2E_{f_p}[p^c(1 - p)^c])^n,$$

which follows after assuming that the components of $\mathbf{p}$ are iid r.v.'s distributed according to f_p. Observe that this expectation is maximized simply by considering a pmf that takes 1 on the maximum of the argument of the expectation and 0 otherwise. Since $p^c(1-p)^c$ is symmetric around $1/2$, the expected value is maximized simply by taking $p = 1/2$ with probability 1. □

The previous lemma suggests that codes with approximately the same number of zeros and ones in each row of the codebook are good candidates to be (c, c)-separating codes. Equivalently, for each set of $2c$ rows of the codebook, one would expect that every possible $\mathbb{F}_2^{2c}$-configuration exhibit a uniform distribution approximately. In fact, there exist constructions of (c, c)-separating codes which are based on this observation [61].

5.2.2 Universal and Almost Universal Sets

Universal sets have been described in Definition 5.7. Moreover, it has been shown that the construction of universal sets can be reduced to the construction of ε-biased arrays.

It is easy to see that an $(M, 2c)$-universal set of size n also yields a (c, c)-separating (n, M)-code [61]. To see this, let A be an (n, M)-array whose rows form an $(M, 2c)$-universal set. Now, regard the columns of A as the codewords of a code C. Consider two disjoint c-subsets $U, V \subseteq C$, i.e., $2c$ columns of A. Since the rows of A are an $(M, 2c)$-universal set, this means that for the selected $2c$ columns every possible $\mathbb{F}_2^{2c}$-configuration appears. In particular, there is a row i where all the columns corresponding to U contain symbol 0 and all the columns corresponding to V contain symbol 1 in that particular row. Hence i is a separating position for coalitions U, V, i.e., $P_i(U) \cap P_i(V) = \emptyset$, as desired. Recall again that this is the same as a c-secure frameproof code when we are talking about absolute separation.

Efficient constructions of $(M, 2c)$-universal sets using ε-biased from $2c$-wise independence arrays are presented in [57], by virtue of Proposition 5.8 and Corollary 5.9. These constructions yield a (c, c)-separating code of length $\log_2 M \cdot 2^{O(2c)}$. Using this idea, we aim to relax the constraint imposed by the $(M, 2c)$-universality to obtain a code with a better rate. In fact, we do not need that every possible $\mathbb{F}_2^{2c}$-configuration appears in the code, for every choice of $2c$ codewords. Hence, we propose to relax Definition 5.7 by allowing a fraction of vectors $\mathbf{a} \in \mathbb{F}_2^{2c}$, not to appear in the projection on a subset $S \subseteq \{1, \ldots, M\}$ of $2c$ positions. This is formalized in the following definition.

Definition 5.11. An *ε-almost (M,t)-universal set* B is a subset of $\mathbb{F}_2^M$ such that for every subset $S \subseteq \{1,\ldots,M\}$ of t positions the set of projections of the elements of B on the indices of S contains a fraction of $1-\varepsilon$ or more $\mathbb{F}_2^t$-configurations.

Again, if A is a binary (n,M)-array, the rows of A generate an ε-almost (M,t)-universal set provided that there are at least $2^t(1-\varepsilon)$ vectors $\mathbf{a} \in \mathbb{F}_2^t$ such that $\nu_S(\mathbf{a};A) > 0$, for every subset $S \subseteq \{1,\ldots,M\}$ of t columns.

Similarly as Proposition 5.8, the following results show the connection between ε-almost (M,t)-universal sets and ε-away from t-wise independence arrays.

Proposition 5.12. Let A be a binary (n,M)-array A. If A is $(\varepsilon + 2^{-t})$-away from t-wise independence, then the rows of A yield an ε-almost (M,t)-universal set of size n.

Proof. Assume by contradiction that the rows of A do not yield an ε-almost (M,t)-universal set. In other words, there is a subset $S \subseteq \{1,\ldots,M\}$ of t columns such that there are strictly more than $2^t\varepsilon$ configurations $\mathbf{a} \in \mathbb{F}_2^t$ such that $\nu_S(\mathbf{a};A) = 0$. For this particular subset S we have

$$\begin{aligned}
&\sum_{\mathbf{a}\in\mathbb{F}_2^t} |n^{-1}\nu_S(\mathbf{a};A) - 2^{-t}| \\
&\quad \geq (\lfloor 2^t\varepsilon \rfloor + 1)2^{-t} + \sum_{\substack{\mathbf{a}\in\mathbb{F}_2^t \text{ s.t.} \\ \nu_S(\mathbf{a};A)>0}} |n^{-1}\nu_S(\mathbf{a};A) - 2^{-t}| \\
&\quad \overset{\text{(a)}}{\geq} (\lfloor 2^t\varepsilon \rfloor + 1)2^{-t} + 1 - 2^{-t}(2^t - \lfloor 2^t\varepsilon \rfloor - 1) = (\lfloor 2^t\varepsilon \rfloor + 1)2^{-t+1}.
\end{aligned}$$

Inequality (a) follows after applying the Pareto optimality criterion for resource allocation with additive convex objective. It is routine to check that

$$(\lfloor 2^t\varepsilon \rfloor + 1)2^{-t+1} > \varepsilon + 2^{-t}$$

for all $\varepsilon \geq 0$. This contradicts the fact that A is $(\varepsilon + 2^{-t})$-away from t-wise independence. □

5.2.3 Construction of Almost Universal Sets

As Proposition 5.12 states, the construction of an ε-almost (M, t)-universal set reduces to constructing an $(\varepsilon+2^{-t})$-away from t-wise independence array, and by Corollary 5.9, it reduces to the construction of a weakly biased array. Moreover, it is easy to see that the array A from Corollary 5.9 can be regarded as a t-wise ε-biased array, which is a less restrictive condition than an ε-biased array.

A standard construction of t-wise ε-biased binary arrays is also presented in [57].

Theorem 5.13. Let A be an ε-biased binary (n, M')-array, and let H be the parity-check matrix of a binary $[M, M - M']$-code with minimum distance $t + 1$. Then, the matrix product $A \times H$ is a t-wise ε-biased (n, M)-array.

Usually, the matrix H used in Theorem 5.13 above is the parity-check matrix of a binary $[M, M - M']$-BCH code with minimum distance $t + 1$. In this case, the matrix H has M columns and $M' = t \log_2 M$ rows. It is shown in [57] that, by using Theorem 5.13 in Corollary 5.9, the number of rows of an ε-away from t-wise independence (n, M)-array can be reduced from $n = 2^{O(t+\log M+\log \frac{1}{\varepsilon})}$ to $n = 2^{O(t+\log\log M+\log \frac{1}{\varepsilon})}$.

The problem now reduces to obtain binary ε-biased (n, M)-arrays with n as small as possible. From [58], we have the following result.

Theorem 5.14. There exists an explicit construction of a binary (n, M)-array that is ε-biased, with

$$n \leq 2^{2(\log_2 M+\log_2 \frac{1}{\varepsilon})}.$$

However, in [56], better explicit construction of ε-biased arrays are given, when the parameters satisfy some required conditions. The best construction shown there is based in Suzuki codes. Below we rewrite [56, Theorem 10] in our notation.

Theorem 5.15. If $\log_2 M > 3 \log_2 \frac{1}{\varepsilon}$, then there exists an explicit construction of a binary (n, M)-array that is ε-biased, with

$$n \leq 2^{3/2\,(\log_2 M+\log_2 \frac{1}{\varepsilon})+2}.$$

Hence, to construct an ε-almost (M,t)-universal set we can proceed as follows.

Construction 5.16. Let M and t be integers and $0 \leq \varepsilon < 1$.

1) Construct an (n, M')-array A' that is ε'-biased, where we take $M' = t \log_2 M$ and $\varepsilon' = 2^{-t/2}(\varepsilon + 2^{-t})$.
2) Construct the parity-check matrix H of a BCH code of length M, codimension $M' = t \log_2 M$ and minimum distance $t+1$.
3) The matrix product $A = A' \times H$ generates a t-wise ε'-biased (n, M)-array.
4) By Corollary 5.9, the array A is also $(\varepsilon + 2^{-t})$-away from t-wise independence.
5) Hence, by Proposition 5.12, the rows of A generate an ε-almost (M,t)-universal set.

Observe that the conditions of Theorem 5.15 apply in Step 1) in the construction above when $\log_2 M' > -3\log_2\big(2^{-t/2}(\varepsilon + 2^{-t})\big)$, i.e.,

$$\log_2 t + \log_2 \log_2 M > 3\,t/2 - 3\log_2(\varepsilon + 2^{-t}).$$

The resulting ε-almost (M,t)-universal set, using Theorem 5.15, has size

$$n \leq 2^{3/2(t/2+\log_2 t+\log_2 \log_2 M-\log_2(\varepsilon+2^{-t}))+2}.$$

We remark that the condition above, even though analytically meaningful, it is only satisfied for impractically large values of M. That is, it will lead to codes with an excessively large number of codewords. For practical scenarios, using the constructions for weakly biased arrays given from Theorem 5.14, the resulting ε-almost (M,t)-universal sets have size

$$n \leq 2^{2(t/2+\log_2 t+\log_2 \log_2 M-\log_2(\varepsilon+2^{-t}))}. \tag{5.1}$$

In both cases the length of the construction is $n = \log_2 M \cdot 2^{O(t-\log(\varepsilon+2^{-t}))}$.

We conclude this section with the following result that will be useful below.

Lemma 5.17. Let B be an ε-almost (M,t)-universal set. Then, B is an (M,t')-universal set with $t' = \min\{t, \lceil \log_2 \frac{1}{\varepsilon} \rceil - 1\}$.

Proof. For each subset $S \subseteq \{1, \ldots, M\}$ of t indices, let $z = 2^t\varepsilon$ denote the maximum number of missing $\mathbb{F}_2^t$-configurations. Observe that if $z < 2^{t-t'}$, then B is (M, t')-universal. To see this, note that to remove an $\mathbb{F}_2^{t'}$-configuration we need to remove, at least, $2^{t-t'}$ $\mathbb{F}_2^t$-configurations. Hence, as long as t' is so that the aforementioned condition is satisfied, i.e., $t' < \log_2 \frac{1}{\varepsilon}$, the set B is (M, t')-universal. □

5.2.4 Application to Almost Secure Frameproof Codes

Recall from Section 5.2.2 that for $t = 2c$ an (M, t)-universal set of size n generates a c-secure frameproof (n, M)-code. Now, take an (n, M)-array A whose rows generate an ε'-almost (M, t)-universal set B with $t \geq c$, and regard its columns as the codewords of a code C. Since C is generated from an (n, M)-array A it is an (n, M)-code of rate $R = \log_2 M/n$.

Now, let us focus on the frameproof properties of such a code C. According to Lemma 5.17, for $t \geq 2c$ and $\varepsilon' < 2^{-2c}$, the ε'-almost (M, t)-universal set B is $(M, 2c)$-universal and hence, C is c-secure frameproof, as we have just recalled. If $t < 2c$, or if $t \geq 2c$ and $\varepsilon' \geq 2^{-2c}$, then B is not $(M, 2c)$-universal. However, in this latter case, it could happen that C is still c-secure frameproof. Note that a c-secure frameproof code only needs a separating position for every pair of c-subsets, which is a less strict requirement than $(M, 2c)$-universality. This means that, in order to lose the c-secure frameproof property, we have to remove, at least, a fraction of

$$\varepsilon \geq \frac{2^{t-2c} + 2^{t-2c}}{2^t} = 2^{-2c+1}$$

$\mathbb{F}_2^t$-configurations from each projection of t positions of the (M, t)-universal set.

In the cases just mentioned, the underlying code has to be regarded as an almost secure frameproof code. For technical reasons, we restrict our study to the case $t > c$. For $t < c$, even using (M, t)-universal sets, it is not guaranteed the existence of enough positions where all the codewords of a c-coalition have the same code element, which will be a requirement in the proof below. Also, for $t = c$ an (M, t)-universal set from Construction 5.16 only guarantees the existence of two such positions.

The following proposition formalizes the relationship between ε-almost c-secure frameproof codes and ε'-almost (M,t)-universal sets that we have constructed above.

Proposition 5.18. Let $c \geq 2$, t, M be integers such that $M \geq 2c$, and let one of the following conditions be satisfied

1) $c < t < 2c$ and $0 \leq \varepsilon' < 2^{-c} - 2^{-t}$, or
2) $t \geq 2c$ and $2^{-2c+1} \leq \varepsilon' < 2^{-c} - 2^{-t}$.

Then, an ε'-almost (M,t)-universal set of size n from Construction 5.16 generates an ε-almost c-secure frameproof (n,M)-code, for

$$\varepsilon \geq M^c(1 - 2^{-c} + 2^{-t} + \varepsilon')^n. \tag{5.2}$$

Proof. Consider a code C generated from an ε'-almost (M,t)-universal set B, as stated. By virtue of Lemma 5.17, B is an (M,c)-universal set when either condition is satisfied. Let A be the (n,M)-array used to construct B, which, according to Proposition 5.12, is $(2^{-t}+\varepsilon')$-away from t-wise independence. Moreover, as noted in Remark 5.6, it is also a t-wise $(2^{-t}+\varepsilon')$-dependent array. This means that every $\mathbb{F}_2^c$-configuration appears in every subset of c columns with probability p satisfying

$$2^{-c} - 2^{-t} - \varepsilon' \leq p \leq 2^{-c} + 2^{-t} + \varepsilon'.$$

In other words, the codewords of a random c-coalition from C have the same symbol in a given position with probability p.

Now, we can operate similarly as in Theorem 4.11. Let $\mathbf{z}$ be a descendant generated by some c-coalition of the code, $\mathbf{z} \subseteq \mathrm{desc}_c(C)$. The probability that $\mathbf{z}$ belongs to another c-coalition V is at most $(1-p)^n$. Indeed, for every position $1 \leq i \leq n$, the probability $\Pr\{z_i \in P_i(V)\}$ is $\leq 1-p$. Hence, by using the union bound, we can bound the probability that $\mathbf{z}$ is generated by some other coalition of the code as $\leq M^c(1-p)^n$. The ratio (probability) of not uniquely decodable descendants in $\mathrm{desc}_c(C)$ is therefore $\leq \varepsilon$, which means that C is an ε-almost c-secure frameproof (n,M)-code. □

Remark 5.19. Observe that in Proposition 5.18 above there is the requirement that the ε'-almost (M,t)-universal set be generated from an $(2^{-t}+\varepsilon')$-away from t-wise independence array. If this is not the case, for $t > c$, an arbitrary ε'-almost (M,t)-universal set only guarantees that every $\mathbb{F}_2^c$-configuration is repeated at most 2^{t-c} times. Consequently, each c-coalition is guaranteed to have only 2^{t-c} constant-valued positions. This would yield an ε-almost c-secure frameproof code with $\varepsilon \geq M^c\, 2^{-2^{t-c}}$, which is of impractical use.

In order to ease the analysis, one could assume that for every subset of at most c indices, each possible $\mathbb{F}_2^t$-configuration appears with uniform probability in the (M,c)-universal sets in the proof above, obtaining ε-almost c-secure frameproof codes for $\varepsilon \geq M^c(1-2^{-c})^n$. This is a reasonable assumption, since universal sets generated from weakly biased arrays are indeed "almost uniform" probability sample spaces. However, the error probability from Proposition 5.18 is already negligible, and this assumption would not handle the case $t = c$ properly.

5.2.5 Results for Some Coalition Sizes

In Table 5.1 we show the computed code rates for ε-secure frameproof codes from Proposition 5.18, for the case of coalitions of size $c = 2$ and 3. We are considering ε'-almost (M,t)-universal sets with $t = 2c$ and, at most, $z = 2^t\varepsilon'$ missing $\mathbb{F}_2^t$-configurations. Recall that when $\varepsilon' < 2^{-2c+1}$, i.e., $z < 2$ in this example, the code is (c,c)-separating, that is $\varepsilon = 0$. The value of ε provided in the table corresponds to the worst-case for every given row. The code rates have been computed for code sizes of $M = 10^3$, 10^4, 10^5, 10^6 and 10^7 users, using the constructions of almost universal sets derived from weakly biased arrays constructed according to Theorem 5.14. Note how the code rate increases significatively as $z = 2^t\varepsilon'$ increases. For example, for $c = 2$, we can obtain almost 2-secure frameproof codes with small error and with a rate 10 times higher than that of ordinary $(2,2)$-separating codes constructed according to [61] (equivalent to the first row of Table 5.1).

c	z	$\log_{10}\varepsilon$	Code size				
			$M = 10^3$	$M = 10^4$	$M = 10^5$	$M = 10^6$	$M = 10^7$
2	0	n/a	$1.531 \cdot 10^{-6}$	$1.148 \cdot 10^{-6}$	$9.187 \cdot 10^{-7}$	$7.656 \cdot 10^{-7}$	$6.562 \cdot 10^{-7}$
2	1	n/a	$6.124 \cdot 10^{-6}$	$4.593 \cdot 10^{-6}$	$3.675 \cdot 10^{-6}$	$3.062 \cdot 10^{-6}$	$2.625 \cdot 10^{-6}$
2	2	$-2.26 \cdot 10^4$	$1.378 \cdot 10^{-5}$	$1.034 \cdot 10^{-5}$	$8.268 \cdot 10^{-6}$	$6.890 \cdot 10^{-6}$	$5.906 \cdot 10^{-6}$
3	0	n/a	$1.063 \cdot 10^{-8}$	$7.975 \cdot 10^{-9}$	$6.380 \cdot 10^{-9}$	$5.316 \cdot 10^{-9}$	$4.557 \cdot 10^{-9}$
3	1	n/a	$4.253 \cdot 10^{-8}$	$3.190 \cdot 10^{-8}$	$2.552 \cdot 10^{-8}$	$2.127 \cdot 10^{-8}$	$1.823 \cdot 10^{-8}$
3	2	$-3.79 \cdot 10^6$	$9.569 \cdot 10^{-8}$	$7.177 \cdot 10^{-8}$	$5.742 \cdot 10^{-8}$	$4.785 \cdot 10^{-8}$	$4.101 \cdot 10^{-8}$
3	3	$-1.64 \cdot 10^6$	$1.701 \cdot 10^{-7}$	$1.276 \cdot 10^{-7}$	$1.021 \cdot 10^{-7}$	$8.506 \cdot 10^{-8}$	$7.291 \cdot 10^{-8}$
3	4	$-7.81 \cdot 10^5$	$2.658 \cdot 10^{-7}$	$1.994 \cdot 10^{-7}$	$1.595 \cdot 10^{-7}$	$1.329 \cdot 10^{-7}$	$1.139 \cdot 10^{-7}$
3	5	$-3.59 \cdot 10^5$	$3.828 \cdot 10^{-7}$	$2.871 \cdot 10^{-7}$	$2.297 \cdot 10^{-7}$	$1.914 \cdot 10^{-7}$	$1.640 \cdot 10^{-7}$
3	6	$-1.31 \cdot 10^5$	$5.210 \cdot 10^{-7}$	$3.908 \cdot 10^{-7}$	$3.126 \cdot 10^{-7}$	$2.605 \cdot 10^{-7}$	$2.233 \cdot 10^{-7}$

Table 5.1: Some attainable code rates for explicit constructions of ε-almost c-secure frameproof codes of size between 10^3 and 10^7.

5.2.6 Explicit Constructions of Fingerprinting Codes

Finally, we show how binary ε-almost c-secure frameproof codes can be used to explicitly construct a family of binary fingerprinting codes with an efficient decoding algorithm.

In Chapter 4 existence conditions for a family of concatenated fingerprinting codes is proposed, using a Reed-Solomon as outer code and an almost separating or almost secure frameproof codes as inner code. Note that, from (5.1), the rate R of the binary ε-almost c-secure frameproof (n, M)-codes from Proposition 5.18 attain its maximum value for $t = c + 1$, that is,

$$R \leq 2^{-2(\frac{3}{2}(c+1)+\log_2(c+1))-\log_2\log_2 M}.$$

Hence, combining Corollary 4.16 with the results from this chapter we have the following result.

Corollary 5.20. Let q, c, t, be integers, $q > c^2$, $t > c$. Moreover, let ε_{in} and σ be so that

$$\varepsilon_{\text{in}}^{\min} \leq \varepsilon_{\text{in}} < \sigma < \frac{q - c^2}{q - c},$$

where $\varepsilon_{\text{in}}^{\min}$ depends on q, t and c. Then, for any fixed rate R such that

$$R < \frac{1-\sigma}{c(c+1)} R_{\text{in}}, \qquad \text{with } R_{\text{in}} \leq 2^{-2(\frac{3}{2}(c+1)+\log_2(c+1))-\log_2\log_2 q},$$

there exists an explicit construction of a c-secure with ε-error family of binary codes $\mathcal{C} = \{C_t\}_{t\in T}$ of length n, with polynomial-time identification algorithm, rate R, and error probability ε decreasing exponentially as

$$\varepsilon \leq 2^{-n(\frac{1-\sigma}{c}R_{\text{in}}-(c+1)R+o(1))} + 2^{-nD(\sigma\|\varepsilon_{\text{in}})}.$$

Remark 5.21. The parameter $\varepsilon_{\text{in}}^{\min}$ in the previous corollary takes its value from (5.2). Then, it depends on the parameters q, t and c according to the associated $(2^{-t}+\varepsilon')$-away from t-wise independence array used in Proposition 5.18.

As noted in Chapter 4, the use of almost secure frameproof codes instead of ordinary secure frameproof codes introduces an additional error term in the identification process. Note again that this error term decreases exponentially with the outer code length.

5.3 Conclusion

Almost separating and almost secure frameproof codes are two relaxed versions of separating codes. In this chapter, we have presented explicit constructions of almost secure frameproof codes.

Our work has started with the study of the connection between weakly dependent arrays and universal sets, and the subsequent connection between universal sets and separating codes.

Starting with this idea, we have introduced a relaxation in the definition of a universal set. We show that an almost universal set can be used to construct an almost secure frameproof code. This observation has lead us to the explicit constructions of almost secure frameproof codes presented. We have proposed a construction based on Suzuki codes, which provide one of the best constructions known for weakly biased

arrays. For practical uses, however, we have to switch to the constructions of small-bias probability spaces proposed by Alon et al.

We remark that, as expected, the explicit constructions presented are somewhat far from the theoretical existence bounds shown in earlier works. For example, probabilistic arguments from Chapter 4 show the existence of asymptotically almost 2-secure frameproof families of codes of rate $R = 0.2075$, whereas the explicit constructions that we have presented above provide codes of rate below this figure. Nevertheless, our work shows the existence of constructible almost secure frameproof codes of much higher rate than secure frameproof codes based on weakly biased arrays. Also, the main point of our work is to present the first explicit and practical-use constructions for such families of codes.

We have also shown how the proposed constructions can be used to explicitly construct a family of fingerprinting codes. The construction presented is based on the theoretical existence results, also from Chapter 5, which assumed the existence of almost secure frameproof codes. Hence, another of the main contributions of this chapter has been to provide a "real" implementation of such a theoretical existence result for a fingerprinting scheme. As discussed in Theorem 4.15 and Corollary 4.16, replacing ordinary separating codes by almost secure frameproof codes introduces an additional error term in the identification of guilty users that, fortunately, decreases exponentially with the outer code length.

Finally, we would like to note that even though a universal set is a separating code, the relationship between an almost universal set and an almost separating code is by no means evident and will we the subject of future research.

The results of this chapter have been published in [10].

Chapter 6

The Separating and Traceability Properties of Reed-Solomon Codes

Under the narrow-sense envelope model it is possible to identify traitors with zero-error probability. Recall from Section 2.1 that c-IPP and c-TA codes allow the unambiguous identification of traitors from coalitions of size at most c. The existence conditions for IPP codes are less strict than those for TA codes. Also, as opposed to TA codes, IPP codes do not have an efficient identification algorithm in the general case, i.e., they cannot be decoded using a minimum-distance decoding algorithm. On the other hand, separating codes possess weaker identification capabilities, and do not guarantee unambiguous identification of traitors. It is a well-known result (2.7) that a TA code is an IPP code, and an IPP code is a separating code. The converse is in general false. However, it has been conjectured that for Reed-Solomon codes all three properties are equivalent. In this chapter we investigate this equivalence, providing a positive answer for a large number of cases.

The motivation for the work in this chapter comes from a problem posed by Silverberg et al. in [36, 37], regarding the connection between the IPP and the TA properties of Reed-Solomon codes. However, it is worth noticing here that a more general question was introduced earlier by Sagalovich in [21].

This chapter is organized as follows. In the next section we introduce the topic and present some previous results. In Section 6.2 we present the main results of the

chapter, showing the equivalence of some combinatorial properties for Reed-Solomon codes, when certain conditions are met. Next, in Section 6.3 we provide an illustrative example and a table summarizing the results. Finally we present the conclusions.

6.1 Statement of the Problem

Let us begin this section by introducing some concepts and notation that will be useful in this chapter.

Let C be an (n, M)-code, and let $U, V \subseteq C$ be two (disjoint) subsets of size c and c', respectively. Consider the projections $P_i(U), P_i(V)$ on the ith position as defined in (2.2). Similarly as in [21], let us denote by $\theta(U, V)$ the number of separating positions between U and V, i.e.,

$$\theta(U, V) \stackrel{\text{def}}{=} |\{i : P_i(U) \cap P_i(V) = \emptyset, 1 \leq i \leq n\}|. \tag{6.1}$$

According to the nomenclature introduced in Chapter 4, if $\theta(U, V) = 0$, then the subsets U and V are not separated. Also, for a code C, let us denote $\theta_{c,c'}(C)$ the smallest value $\theta(U, V)$ attained for disjoint subsets $U, V \subseteq C$ of size c and c', respectively. We shall immediately become less formal and we will simply use $\theta_{c,c'}$ when the code under study C is clear from the context. Of course, $\theta_{c,c'} = \theta_{c',c}$, and although in general $\theta(U, V)$ is not a metric in the mathematical sense of the term, clearly, $\theta(\{\mathbf{u}\}, \{\mathbf{v}\}) = d(\mathbf{u}, \mathbf{v})$ and $\theta_{1,1} = d(C)$.

The values $\theta_{c,c'}$ will be useful in the characterization of codes with separating and traceability properties. In fact, a (c, c')-separating code can be defined as a code C that satisfies $\theta_{c,c'} > 0$.

Combining (2.7) with the results from [62] it is easy to see that for a code C

$$\begin{aligned} d(C) > (1 - 1/c^2)n \Rightarrow \theta_{c,1} > (1 - 1/c)n \\ \Rightarrow c\text{-TA} \Rightarrow c\text{-IPP} \Rightarrow (c, c)\text{-separating}. \end{aligned} \tag{6.2}$$

6.1.1 The Separating and Traceability Properties in MDS Codes

The Singleton bound states that for an (n, M)-code with minimum distance d, we have $M \leq q^{n-d+1}$. Codes that achieve equality in the Singleton bound are called *maximum distance separable* (MDS) codes. Therefore, linear MDS $[n, k]$-codes have minimum distance $d = n - k + 1$.

Even though the implications in (6.2) are well-known and obvious, it took several years to prove the converse of the first and second implication for linear MDS codes. The next result first appeared in [62].

Theorem 6.1 ([62, Theorem 2.3])**.** Let C be an MDS $[n, k]$-code with minimum distance d over the finite field $\mathbb{F}_q$ such that $n \leq q + 1$. Then, for $c \geq 2$, C is a c-TA code if and only if $d > (1 - 1/c^2)n$.

Putting this together with (6.2), we conclude that if C is a linear MDS $[n, k]$-code, then

$$d(C) > (1 - 1/c^2)n \Leftrightarrow \theta_{1,c} > (1 - 1/c)n \Leftrightarrow c\text{-TA}.$$

A well-known family of linear MDS codes are Reed-Solomon codes [46, 47]. Consider the following definition.

Definition 6.2. Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a subset of n elements of $\mathbb{F}_q$, called *evaluation points*. We define the $[n, k]$-code $\mathrm{G}(n, k)$ over $\mathbb{F}_q$ as

$$\mathrm{G}(n, k) \stackrel{\text{def}}{=} \{(f(\gamma_1), \ldots, f(\gamma_n)) : f(x) \in \mathbb{F}_q[x],\ \deg f(x) < k\}.$$

Note that the code $\mathrm{G}(n, k)$ is a linear MDS code, irrespective of the choice of the set of evaluation points. If Γ is the multiplicative group of the ground field, $\mathbb{F}_q^*$, then $\mathrm{G}(n, k)$ is the $[n, k]$-Reed-Solomon code as described in Definition 3.1. If $\Gamma = \mathbb{F}_q$, then it is known as *extended Reed-Solomon code*.

In [36, 37], the authors posed the following question.

Question 6.3. Is it the case that $d > (1 - 1/c^2)n$ for all c-IPP Reed-Solomon codes of length n and minimum distance d?

In fact, we will see below that, for many families of Reed-Solomon codes, the condition $d \leq (1 - 1/c^2)n$ implies not only losing the c-IPP property, but also losing the (c, c)-separating property. Hence, the converse of all the implications in (6.2) holds for such families.

Let C' and C be $[n, k']$ and $[n, k]$-Reed-Solomon codes, respectively, over $\mathbb{F}_q$. Observe that for $k' \leq k$, we have $C' \subseteq C \subseteq \mathbb{F}_q^n$. Therefore, to provide a positive answer to the question above, we only need to show that the $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$ with $k = \lceil n/c^2 \rceil + 1$ has $\theta_{c,c} = 0$, for every possible pair of values q and c.

Remark 6.4. A possible strategy to tackle the question above can be as follows. If it can be shown that the $[n, k]$-Reed-Solomon code has $\theta_{c,c'} = \max\{0, n - c\,c'(k-1)\}$, then a positive answer to the question above would be immediate. Taking $c = c'$ and $d = n - k + 1 \leq (1 - 1/c^2)n$ would imply $\theta_{c,c} = 0$. This strategy was somehow pointed out in [21].

The remark that we have just made suggests a generalization of Question 6.3 as follows.

Question 6.5. Is it the case that $\theta_{c,c'} = \max\{0, n - c\,c'(k-1)\}$ for all $\mathrm{G}(n, k)$ codes from Definition 6.2?

The motivation for these questions arises from the fact that the amount of information (fingerprint) that we can embed in a digital document is limited. Assume that we can embed no more than n symbols from $\mathbb{F}_q$. Then, there exists a c-TA Reed-Solomon code that can allocate q^k users, for any $k < n/c^2 + 1$. If for the same value of n the distributor needs to allocate more users, then by Theorem 6.1 the code will not be c-TA. In this situation, is there a chance that we can still identify traitors? The remark made above suggests that for $k \geq n/c^2 + 1$ there are neither c-IPP nor (c, c)-separating codes, hence identification with zero-error probability would not be possible.

In this chapter we are mainly concerned with giving an answer to Question 6.3. However, the constructions presented also provide some answers for Question 6.5.

6.1.2 Previous Results

In connection with Remark 6.4, an answer to the questions above for the case $c = 2$ can be found in [21]. It is written there that in 1986 G. D. Katsman and S. N. Litsyn applied Mattson-Solomon polynomials and linearized polynomials to Reed-Solomon codes obtaining

$$\theta_{2,2} = n - 4(k-1).$$

Taking $k \geq n/4 + 1$, we have $\theta_{2,2} = 0$. Therefore $(2,2)$-separating $\Rightarrow d > (1 - 1/4)n$, which means that the converse of every implication in (6.2) holds for Reed-Solomon codes and the particular case $c = 2$. Unfortunately, the proof of this nice result has not been published.

Also, in [36,37] a custom-made construction of $\mathrm{G}(n,k)$ codes is presented, defined over *sufficiently large* alphabets. They have minimum distance $d = (1 - 1/c^2)n$ and they are not (c,c)-separating. Nevertheless, no specific relation is given between the code parameters.

In [63] a related result is presented for $[n,k]$-Reed-Solomon codes such that their ground field contains the $(k-1)$th roots of unity. The idea there was to restate the separating condition algebraically, as a system of equations. From [63, Theorem 7], and from the proof provided by the authors, the following corollary is immediate.

Corollary 6.6. Let C be an $[n,k]$-Reed-Solomon code over $\mathbb{F}_q$ with minimum distance d. If $n - d$ divides $q - 1$, then C is (c,c)-separating if and only if $d > (1 - 1/c^2)n$.

6.2 Equivalence of the Separating and Traceability Properties of Reed-Solomon Codes

We begin by showing some upper and lower bounds of $\theta_{c,c'}$ for linear and MDS codes. These bounds were presented for the particular cases $c = c' = 2$ in [21], and $c = 1$, c' arbitrary in [62].

Lemma 6.7. Let C be an $[n,k]$-code with minimum distance $d = d(C)$, and let c, c' be two positive integers. Then,

$$\max\{0, d-(c\,c'-1)(n-d)\} \leq \theta_{c,c'} \leq \max\{0, d-(c+c'-2)(k-1)\}. \tag{6.3}$$

If C is additionally an MDS code and $c, c' \geq 2$, then

$$\begin{aligned}\max\{0, d-(c\,c'-1)(n-d)\} &\leq \theta_{c,c'} \\ &\leq \max\{0, d-(c+c'-2)(k-1)-c-c'+3\}. \end{aligned} \tag{6.4}$$

Proof. Let U, V be any two disjoint subsets of C of size c and c', respectively. Note that two different codewords of C agree in at most $n-d$ positions. Also, from (6.1), the number of positions i such that $P_i(U) \cap P_i(V) \neq \emptyset$, i.e., the number of nonseparating positions, is $n - \theta_{c,c'}$. Hence, for every codeword $\mathbf{u} \in U$, the codewords in V can match together at most $c'(n-d)$ positions of $\mathbf{u}$. Since U has c elements, we have $n - \theta_{c,c'} \leq c\,c'(n-d)$, which proves the lower bounds in (6.3) and (6.4).

To prove the upper bounds, construct two subsets U and V in the following way. First, take any $\mathbf{u}, \mathbf{v} \in C$ such that $d(\mathbf{u}, \mathbf{v}) = d$. Such codewords exist, by definition of the minimum distance. Put $\mathbf{u}$ into U and $\mathbf{v}$ into V. Now insert $c-1$ codewords in U such that each one matches $k-1$ disjoint positions of $\mathbf{v}$, where $\mathbf{u}$ and $\mathbf{v}$ differ. Such $c-1$ codewords exist by virtue of [62, Lemma 2.2]. Equivalently, insert $c'-1$ codewords in V such that each one matches $k-1$ disjoint positions of $\mathbf{u}$, where $\mathbf{u}$ and $\mathbf{v}$ differ. Therefore, the number of positions i such that $P_i(U) \cap P_i(V) \neq \emptyset$, i.e., where the elements of U and V have a common element, is $n-d+(c+c'-2)(k-1)$, which proves the upper bound in (6.3).

Recall that in an MDS code we can regard any k positions as information positions. Hence, for an MDS code and $c, c' \geq 2$, we can force an additional position of every codeword of $V \setminus \{\mathbf{v}\}$ to match a position of a given codeword $\mathbf{u}' \in U \setminus \{\mathbf{u}\}$. Similarly, we can set an additional position of each codeword of $U \setminus \{\mathbf{u}, \mathbf{u}'\}$ to match a position of any other codeword in $V \setminus \{\mathbf{v}\}$. This reduces the number of nonseparating positions in $c+c'-3$, and proves the upper bound in (6.4). □

Consider the case $c = c'$. For linear MDS $[n, k]$-codes with minimum distance d, and from the previous lemma, it is clear that when $d - 2(c-1)(k-1) - 2c + 3 \leq 0$, we have $\theta_{c,c} = 0$. Therefore the code is not (c, c)-separating. Also, when we have $d - (c^2 - 1)(n - d) > 0$, then $\theta_{c,c} > 0$ and the code is (c, c)-separating. In fact, the latter condition implies that the code is c-TA. In conclusion, there is an "uncertainty interval," in terms of d, in which the (c, c)-separating property remains to be characterized, namely

$$\frac{2(c-1)n + 2c - 3}{2c - 1} < d \leq (1 - 1/c^2)n.$$

6.2.1 Codes with Multiplicative Subgroups in the Ground Field

Whenever the set of evaluation points Γ is a multiplicative subgroup with generator element α, the code $\mathrm{G}(n, k)$ is (linearly equivalent to) a cyclic code. We denote by $\mathbf{u}^{(i)}$ the cyclic rotation of $\mathbf{u} \in \mathbb{F}_q^n$ in i positions to the right. In this case, it is easy to see that if the polynomial $f(x)$ generates the codeword $\mathbf{u} \in \mathrm{G}(n, k)$, the polynomial $f(\alpha^{-i}x)$ generates the codeword $\mathbf{u}^{(i)}$.

The following result, together with (6.2), generalizes Corollary 6.6 for any $\mathrm{G}(n, k)$ code generated with a multiplicative subgroup of evaluation points, in particular it is valid for Reed-Solomon codes.

Proposition 6.8. Let Γ be a multiplicative subgroup of $\mathbb{F}_q^*$. Also, let $\mathrm{G}(n, k)$ be the code from Definition 6.2, generated with the set of evaluation points Γ, with minimum distance d. If $n - d$ divides n and $d \leq (1 - 1/c^2)n$, then the code is not (c, c)-separating.

Proof. We need to show that under the conditions stated the code contains a non-separated pair of subsets U, V each of size at most c.

Denote $r = n/(k-1)$, and consider the polynomial

$$f(x) = \prod_{i=0}^{k-2} (\alpha^{-ir}x - 1),$$

where α is a generator of Γ. Note that $f(x)$ is a polynomial of degree $k-1$. Hence, the codeword generated from $f(x)$, say $\mathbf{u}$, is in $\mathrm{G}(n,k)$. It is easy to see that $f(\alpha^{-rh}x) = f(x)$ for any integer h. Hence, $\mathbf{u}^{(rh)} = \mathbf{u}$. This, together with the fact that the polynomial has degree $k-1$, means that the codeword $\mathbf{u}$ consists of $k-1$ concatenations of a vector of r distinct elements, say $\mathbf{b} = (b_1, \ldots, b_r)$. Now take $c' = \min\{c, r\} \leq c$ and construct the following set of codewords:

$$U = \{\mathbf{u}^{(ic')} : 0 \leq i < \lceil r/c' \rceil\}.$$

From the starting assumptions, $n - d = k - 1 \geq n/c^2$, which implies that we have $|U| = \lceil r/c' \rceil \leq c' \leq c$. Since $\mathbf{u}$ is the repeated concatenation of the vector $\mathbf{b}$, of length r, and $c'\lceil r/c' \rceil \geq r$, it is clear that for all $1 \leq j \leq n$, there exists a codeword $\mathbf{u}^{(ic')}$ in U such that $u_j^{(ic')} \in \{b_1, \ldots, b_{c'}\}$.

The code $\mathrm{G}(n,k)$ contains all the constant codewords in $\mathbb{F}_q^n$, hence one can construct the set

$$V = \{(b_i, \ldots, b_i) : 1 \leq i \leq c'\},$$

of size $c' \leq c$, which is disjoint from V. Since for U and V every position is not separating, then $\theta(U,V) = 0$. It follows that the code is not (c,c)-separating. □

Corollary 6.9. Let C be an $[n,k]$-Reed-Solomon code over $\mathbb{F}_q$ with minimum distance $d = d(C)$. If $c \geq \sqrt{q-1}$ and $d \leq (1 - 1/c^2)n$, then C is not (c,c)-separating.

Proof. From Proposition 6.8, if $k \geq \lceil n/c^2 \rceil + 1$ and $\lceil n/c^2 \rceil$ divides n, then $\mathrm{G}(n,k)$ is not (c,c)-separating. Reed-Solomon codes have $n = q-1$. Taking $c \geq \sqrt{q-1}$, we have $\lceil n/c^2 \rceil = 1$, and the proof follows. □

It is well-known [36, 37] that c-IPP codes over $\mathbb{F}_q$ do not exist for $c \geq q$. The previous corollary gives a tighter bound for the case of Reed-Solomon codes.

6.2.2 Coalition Size Dividing the Ground Field Size

This section contains the main result of the chapter, which comes in the form of the following theorem.

Theorem 6.10. Let C be an $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$ with minimum distance $d = d(C)$. If c divides q and $d \leq (1 - 1/c^2)n$, then C is not (c, c)-separating.

In fact, from the proof of the theorem, one can easily see that it is valid for any code $\mathrm{G}(n, k)$ with an arbitrary set of evaluation points Γ of size $q - c^2 < |\Gamma| \leq q$.

The proof is based on a special class of polynomials known as linearized polynomials.

Definition 6.11. A polynomial of the form

$$L(x) = \sum_{i=0}^{h} l_i x^{q^i},$$

with coefficients l_i in an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ is called a *linearized polynomial* over $\mathbb{F}_{q^m}$.

Let us present some important, well-known facts [64] about linearized polynomials. First, if $L(x)$ is a linearized polynomial over $\mathbb{F}_{q^m}$, then

$$L(a\alpha + b\beta) = aL(\alpha) + bL(\beta), \tag{6.5}$$

for all $\alpha, \beta \in \mathbb{F}_{q^m}$ and all $a, b \in \mathbb{F}_q$. Thus, the polynomial function $L : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, defined as $x \mapsto L(x)$, is a linear operator on $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Also, the following result will be useful in our proof below.

Theorem 6.12 ([64, Theorem 3.52])**.** Let S be a vector subspace of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then for any nonnegative integer s, the polynomial

$$L(x) = \prod_{\mu \in S} (x - \mu)^{q^s}$$

is a linearized polynomial over $\mathbb{F}_{q^m}$.

For our purposes, we will deal with linearized polynomials over $\mathbb{F}_q = \mathbb{F}_{p^m}$ such that their roots also lie in $\mathbb{F}_q$. We are now in the position to prove Theorem 6.10.

Proof of Theorem 6.10. We prove the theorem by finding a pair of nonseparated c-subsets again.

If $c^2 > q$, the code is not (c, c)-separating by Corollary 6.9. Henceforth, we shall assume that $c^2 \leq q = p^m$. This, together with the fact that c divides q, implies that c^2 also divides q, i.e., $c = p^r$ for some $r \leq m/2$. For any n such that $q - c^2 < n \leq q$, and from the fact that $d \leq (1 - 1/c^2)n$, we conclude that the code contains, at least, all the codewords generated from polynomials of any degree up to $q/c^2 = p^{m-2r}$.

Now, consider the polynomial

$$L(x) = \prod_{\mu \in S} (x - \mu),$$

where S is a vector subspace of $\mathbb{F}_q$ over $\mathbb{F}_p$ of dimension $m - 2r$ and size q/c^2. Note that $L(x)$ is a linearized polynomial by Theorem 6.12. Also, from (6.5) and the fundamental theorem on homomorphisms, the polynomial function $L : \mathbb{F}_q \to \mathbb{F}_q$ is an homomorphism with $|\ker L| = q/c^2$ and $|\operatorname{im} L| = c^2$. Clearly, $\operatorname{im} L$ is a vector subspace of $\mathbb{F}_q$ of dimension $2r$.

Now, take a vector subspace $B \subseteq \operatorname{im} L$ of dimension r and size c. Regard B as an additive subgroup of $\mathbb{F}_q$ and consider its c cosets, which partition $\operatorname{im} L$:

$$B_i = \beta_i + B, \quad 1 \leq i \leq c.$$

We can assume without loss of generality that $\beta_1 = 0$. Now consider the following c polynomials

$$f_i(x) = L(x) - \beta_i, \quad 1 \leq i \leq c.$$

Observe that for every $\gamma \in \mathbb{F}_q$ there is exactly one polynomial $f_i(x)$ with $f_i(\gamma) \in B$. To see this, note that if $L(\gamma)$ lies in the coset B_i of $\operatorname{im} L$, i.e., $L(\gamma) = \beta_i + b$ for some $b \in B$, then the polynomial $f_i(\gamma) = L(\gamma) - \beta_i = \beta_i + b - \beta_i$ evaluates to $b \in B$. The fact that the c cosets B_i partition $\operatorname{im} L$ into disjoint subsets implies that there is only one $f_i(x)$ satisfying this condition.

Now, consider the set of codewords

$$U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\},$$

where $\mathbf{u}^i$ is the codeword generated from the polynomial $f_i(x)$, and the set of c constant codewords

$$V = \{(b, \ldots, b) : b \in B\}.$$

Obviously, U and V are disjoint, because $\deg f_i(x) \geq 1$. Also, $\theta(U, V) = 0$, which proves that the code is not (c, c)-separating.

This construction applies whenever the code contains, at least, all the codewords generated from polynomials of degree up to q/c^2. Since $k - 1 \geq (q-1)/c^2$, this happens in particular for the Reed-Solomon code. Finally, we remark that one can choose an arbitrary coset $\beta_i + B$ for the generation of the constant codewords of the set V. □

However, there are other families of Reed-Solomon codes that can benefit from the constructions presented in the previous proof.

Proposition 6.13. Let C be an $[n, k]$-Reed-Solomon code over $\mathbb{F}_q$ with minimum distance $d = d(C)$. If

$$c' = \sqrt{\frac{q}{\lceil q/c^2 \rceil}} \tag{6.6}$$

is an integer and $d \leq (1 - 1/c^2)n$, then the code is not (c, c)-separating.

Proof. Note that the code contains codewords generated from polynomials of degree at least $\lceil q/c^2 \rceil = q/c'^2$. Also, c' and c'^2 must divide q, which is implied by (6.6). Using the construction from the proof of Theorem 6.10, one can easily see that the code is not (c', c')-separating. The proof follows by noting that $c' \leq c$. □

6.2.3 Summary of Results for Reed-Solomon Codes

We summarize here the results shown in the chapter for the case of $[n, k]$-Reed-Solomon codes with minimum distance $d = n - k + 1$.

1) For any
$$d \leq \frac{2(c-1)n + 2c - 3}{2c-1}$$
the code is not (c, c)-separating.

2) The implication
$$d > (1 - 1/c^2)n \Leftrightarrow (c, c)\text{-separating}$$
is true for families of Reed-Solomon codes when any of the following situations occurs: (a) $c = 2$; (b) $c^2 > q$; (c) $k - 1$ divides n; and (d) $\sqrt{q/\lceil q/c^2 \rceil}$ is an integer value.

Illustratively, in Table 6.1 we show some families of Reed-Solomon codes, for certain values of c and q, satisfying $d > (1 - 1/c^2)n \Leftrightarrow (c, c)$-separating, i.e., of

	$q = 64$	81	125	128	243	256	512	625	729	1024	2187
$c = 2$	a	a	a	a	a	a	a	a	a	a	a
3	c	d	–	–	d	–	–	–	d	–	d
4	d	c	–	d	–	d	d	c	–	d	–
5	c	c	d	–	–	–	–	d	–	–	–
8	d	c	c	d	–	d	d	–	–	d	–
9	b	d	c	d	d	d	c	c	d	–	d
10	b	b	c	d	d	c	–	–	c	c	–
11	b	b	c	d	d	c	–	c	c	–	–
14-15	b	b	b	b	c	–	–	c	c	–	–
16	b	b	b	b	b	d	d	c	–	d	–
17-18	b	b	b	b	b	b	d	c	–	d	–
19	b	b	b	b	b	b	d	c	–	c	–
20-22	b	b	b	b	b	b	d	c	c	c	–
23-24	b	b	b	b	b	b	b	c	c	–	–
25	b	b	b	b	b	b	b	d	c	–	–
26	b	b	b	b	b	b	b	b	c	–	–
27	b	b	b	b	b	b	b	b	d	–	d
28-31	b	b	b	b	b	b	b	b	b	–	d
32	b	b	b	b	b	b	b	b	b	d	d
33	b	b	b	b	b	b	b	b	b	b	d
34-46	b	b	b	b	b	b	b	b	b	b	c
≥ 47	b	b	b	b	b	b	b	b	b	b	b

Table 6.1: Some known families of $[n, k]$-Reed-Solomon codes with $n = q - 1$, $k = \lceil n/c^2 + 1 \rceil$ and minimum distance $d > (1 - 1/c^2)n \Leftrightarrow (c, c)$-separating: (a) $c = 2$; (b) $c^2 > q$; (c) $k - 1$ divides n; and (d) $\sqrt{q/\lceil q/c^2 \rceil}$ is an integer value.

dimension $k = \lceil (q-1)/c^2+1 \rceil$. This, together with several computer-assisted searches, suggests a positive answer to Question 6.3.

6.3 Example

Let us illustrate the proof of Theorem 6.10 with the following example. Consider the finite field $\mathbb{F}_{27} = \mathbb{F}_3[x]/(x^3+2x+1)$ with primitive element $\alpha = \overline{x}$. Let $c = 3$ and take the $[n,k]$-Reed-Solomon code with $n = 26$ and $k = 4$. First, we take the subgroup (or vector space over $\mathbb{F}_3$) $S = \{0, 1, \alpha^{13}\}$ and construct the linearized polynomial

$$L(x) = (x-0)(x-1)(x-\alpha^{13}) = x^3 + \alpha^{13}x.$$

The codeword generated from $L(x)$ is

$$(0, \alpha^{13}, \alpha^9, \alpha^{13}, \alpha^3, \alpha^{16}, \alpha, \alpha^3, \alpha^{22}, \alpha^{13}, \alpha, \alpha, \alpha^9, \\ 0, 1, \alpha^{22}, 1, \alpha^{16}, \alpha^3, \alpha^{14}, \alpha^{16}, \alpha^9, 1, \alpha^{14}, \alpha^{14}, \alpha^{22}),$$

where it can be read that $\operatorname{im} L = \{0, 1, \alpha, \alpha^3, \alpha^9, \alpha^{13}, \alpha^{14}, \alpha^{16}, \alpha^{22}\}$. Since $c^2 = |\operatorname{im} L|$, we take for example the subgroup $B = \{0, 1, \alpha^{13}\} \leq \operatorname{im} L$ of c elements and its c cosets:

$$\begin{aligned} B_1 &= \beta_1 + B = \{0, 1, \alpha^{13}\}, \\ B_2 &= \beta_2 + B = \{\alpha, \alpha^3, \alpha^9\}, \\ B_3 &= \beta_3 + B = \{\alpha^{14}, \alpha^{16}, \alpha^{22}\}, \end{aligned}$$

where $\beta_1 = 0$, $\beta_2 = \alpha$ and $\beta_3 = \alpha^{14}$. Now consider the polynomials $f_i(x) = L(x) - \beta_i$, for $1 \leq i \leq c$. Due to space constraints, we will only show the first 16 positions of their corresponding codewords, which are

$$\begin{aligned} &(0 \ , \alpha^{13}, \boldsymbol{\alpha^9}, \alpha^{13}, \boldsymbol{\alpha^3}, \alpha^{16}, \boldsymbol{\alpha} \ , \boldsymbol{\alpha^3}, \alpha^{22}, \alpha^{13}, \boldsymbol{\alpha} \ , \boldsymbol{\alpha} \ , \boldsymbol{\alpha^9}, 0 \ , 1 \ , \alpha^{22}, \dots), \\ &(\alpha^{14}, \alpha^{22}, 1 \ , \alpha^{22}, \alpha^{13}, \boldsymbol{\alpha^9}, 0 \ , \alpha^{13}, \boldsymbol{\alpha^3}, \alpha^{22}, 0 \ , 0 \ , 1 \ , \alpha^{14}, \alpha^{16}, \boldsymbol{\alpha^3}, \dots), \\ &(\boldsymbol{\alpha} \ , \boldsymbol{\alpha^3}, \alpha^{16}, \boldsymbol{\alpha^3}, \alpha^{22}, 1 \ , \alpha^{14}, \alpha^{22}, \alpha^{13}, \boldsymbol{\alpha^3}, \alpha^{14}, \alpha^{14}, \alpha^{16}, \boldsymbol{\alpha} \ , \boldsymbol{\alpha^9}, \alpha^{13}, \dots), \end{aligned}$$

where the elements of the coset B_2 have been highlighted. These codewords constitute the set U from the proof of Theorem 6.10. Now, call V the set formed by the constant codewords

$$(\boldsymbol{\alpha}\ ,\alpha\ ,\alpha\ ,\alpha\ ,\alpha\ ,\alpha\ ,\boldsymbol{\alpha}\ ,\alpha\ ,\alpha\ ,\alpha\ ,\boldsymbol{\alpha}\ ,\boldsymbol{\alpha}\ ,\alpha\ ,\boldsymbol{\alpha}\ ,\alpha\ ,\alpha\ ,\dots),$$
$$(\alpha^3\ ,\boldsymbol{\alpha^3},\alpha^3\ ,\boldsymbol{\alpha^3},\boldsymbol{\alpha^3},\alpha^3\ ,\alpha^3\ ,\boldsymbol{\alpha^3},\boldsymbol{\alpha^3},\boldsymbol{\alpha^3},\alpha^3\ ,\alpha^3\ ,\alpha^3\ ,\alpha^3\ ,\alpha^3\ ,\boldsymbol{\alpha^3},\dots),$$
$$(\alpha^9\ ,\alpha^9\ ,\boldsymbol{\alpha^9},\alpha^9\ ,\alpha^9\ ,\boldsymbol{\alpha^9},\alpha^9\ ,\alpha^9\ ,\alpha^9\ ,\alpha^9\ ,\alpha^9\ ,\boldsymbol{\alpha^9},\alpha^9\ ,\boldsymbol{\alpha^9},\alpha^9\ ,\dots),$$

that is, the codewords with constant elements in the coset B_2. As we have just shown, for every position, one of the three codewords in U has an element from B_2. Hence, $\theta(U,V)=0$, and both coalitions can generate the same descendant:

$$(\alpha\ ,\alpha^3\ ,\alpha^9\ ,\alpha^3\ ,\alpha^3\ ,\alpha^9\ ,\alpha\ ,\alpha^3\ ,\alpha^3\ ,\alpha^3\ ,\alpha\ ,\alpha\ ,\alpha^9\ ,\alpha\ ,\alpha^9\ ,\alpha^3\ ,\dots).$$

Note that if we had considered the extended Reed-Solomon code instead, it would also have had $k=4$, and the additional position involved would also have satisfied the same property.

6.4 Conclusion

In this chapter we have discussed the separating and traceability properties of Reed-Solomon codes. Our main goal was to give an answer to the question posed by Silverberg et al. in [36, 37]: *Is it the case that $d > (1-1/c^2)n$ for all c-IPP Reed-Solomon codes of length n and minimum distance d?*

We have given a positive answer for some families of Reed-Solomon, when c divides the field size. Also, we have benefited from the proposed constructions to extend the results to other families of punctured Reed-Solomon codes. Obviously this does not provide a full answer to the question but hopefully it gives some hints that may be useful in finding the final response.

The results of this chapter have been published in [1, 2, 8].

Chapter 7

Concluding Remarks

In this dissertation we have addressed several problems that appear in traceability and fingerprinting schemes.

Our contributions from Chapter 3 shows the suitability of the Kötter-Vardy algorithm in a variety of fingerprinting settings. The benefits of using list-decoding for TA codes were already pointed out in [36, 37], and subsequently in [45]. We have shown how the Kötter-Vardy algorithm can be used to identify traitors in TA and IPP Reed-Solomon codes. This algorithm is especially appropriate in these situations, since it eases the reuse of the information obtained in each iteration of the presented algorithms, improving the results obtained in previous works. We have shown how this information can be translated into a reliability matrix in a natural way. Moreover, we have also shown how a family of binary concatenated fingerprinting codes can be constructed in such a way that the use of the Kötter-Vardy algorithm enables polynomial-time identification of traitors in the code length. The presented results extend those from [16] for arbitrary coalition sizes and arbitrary inner codes. Again, we have shown how the use of the Kötter-Vardy algorithm provides a natural framework to deal with the information obtained in the steps of the proposed algorithms.

In Chapter 4 we proposed to relax the ordinary definition of separating code, which is also known under the name of secure frameproof code. The relaxation yielded two different notions, namely, almost separating and almost secure frameproof codes, as opposed to ordinary (absolute) separation, when both notions coincide. The use

of typical sets and probabilistic arguments allowed us show the existence of such codes with better asymptotical rate than that of ordinary separating codes. This fact enables to improve previous constructions of fingerprinting codes, e.g. [15], obtaining codes with better rates preserving exponential decline in the error probability. We have also linked the use of the Kötter-Vardy algorithm to show its applicability in the identification algorithms of the presented codes.

In Chapter 5, we have connected the concept of weakly dependent arrays with the construction of almost secure frameproof codes. Our construction is mainly based in the results presented in [58]. The construction presented is somehow far from the theoretical existence bounds from Chapter 4, however such an explicit construction enables us to connect these results with the previous results to show that explicit constructions of fingerprinting codes based on almost secure frameproof codes exist.

Finally, in Chapter 6 we have given a partial answer to the characterization of IPP Reed-Solomon codes. This question was posed in [36,37]. Our study shows that, in fact, this question has further implications, since it can be seen that the study of IPP/TA Reed-Solomon codes can be linked to the study of the separating property, which is a more "basic" property. We have provided constructive proofs for a large number of families of Reed-Solomon codes, which are also suitable for punctured Reed-Solomon codes. From our main results it seems that a separating and a TA Reed-Solomon code are the same.

7.1 Future Work

Several questions addressed in the present work are subject to future research. These include the following:

– The current definition of the set of TA-parents (Definition 3.8) can be interpreted as the set of IPP-parents (Definition 3.11) that can be efficiently computed according to the algorithms proposed. Is the given definition "tight"? That is, are there more IPP-parents that can be efficiently computed, in polynomial time in the code length? Is it possible to completely characterize and compute this set of parents for other families of codes?

- We have also showed evidence that the almost separating and almost secure frameproof properties from definitions Definition 4.6 and 4.10 are essentially two different notions. Moreover, from our results, we have conjectured that $R_q^{\mathrm{SFP}^*}(c) > R_q^{\mathrm{sep}^*}(c)$. Hence, it would be interesting to find an answer to this question.
- It would be interesting to establish upper bounds on the rate for almost (c, c)-separating and almost c-secure frameproof codes. It seems that establishing tight upper and lower bounds is a rather difficult question, since, even in the simplest case $c = 2$, for ordinary separation the gap between the best upper and lower bounds is significant.
- Universal and almost universal sets have been useful to construct almost secure frameproof codes. Establishing the relationship between an almost universal set and an almost separating code also constitutes another topic of future research. Can "useful" almost separating codes be constructed using almost universal sets?
- It has been already noted that there is strong evidence to think that the separating weight of an $[n, k]$-Reed-Solomon code is $\theta_{c,c'} = \max\{0, n - c\,c'(k-1)\}$, but this has yet to be confirmed. Hence, it would be very interesting to give a complete proof for this question, which would, in turn, give a complete characterization of the separating, IPP and TA properties for Reed-Solomon codes. If proven true, could this result be extended to other families of codes (perhaps MDS codes)?

Bibliography

[1] J. Moreira, M. Fernández, and M. Soriano, "A note on the equivalence of the traceability properties of Reed-Solomon codes for certain coalition sizes," in *Proc. IEEE Int. Workshop Inform. Forensics, Security (WIFS)*, London, United Kingdom, Dec. 2009, pp. 36–40.

[2] J. Moreira, M. Fernández, and M. Soriano, "Propiedades de trazabilidad de los códigos de Reed-Solomon para ciertos tamaños de coalición," in *Proc. Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, Tarragona, Spain, Sep. 2010, pp. 413–417.

[3] M. Fernández, J. Moreira, and M. Soriano, "Identifying traitors using the Koetter-Vardy algorithm," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 692–704, Feb. 2011.

[4] M. Fernández, G. Kabatiansky, and J. Moreira, "Almost separating and almost secure frameproof codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Saint Petersburg, Russia, Aug. 2011, pp. 2696–2700.

[5] J. Moreira, G. Kabatiansky, and M. Fernández, "Lower bounds on almost-separating binary codes," in *Proc. IEEE Int. Workshop Inform. Forensics, Security (WIFS)*, Foz do Iguaçu, Brazil, Nov. 2011, pp. 1–6.

[6] M. Á. Simarro-Haro, J. Moreira, M. Fernández, M. Soriano, A. González, and F. J. Martínez-Zaldívar, "Parallelization of the interpolation process in the Koetter-Vardy soft-decision list decoding algorithm," in *Proc. Int. Conf. Comput. Math. Methods (CMMSE)*, La Manga, Spain, Jul. 2012, pp. 1102–1110.

[7] M. Á. Simarro-Haro, J. Moreira, M. Fernández, M. Soriano, A. González, and F. J. Martínez-Zaldívar, "Paralelización en la interpolación de la decodificación

por listas de códigos Reed-Solomon," in *Proc. Jornadas de Paralelismo (JP)*, Elx, Spain, Sep. 2012.

[8] J. Moreira, M. Fernández, and M. Soriano, "On the relationship between the traceability properties of Reed-Solomon codes," *Adv. Math. Commun.*, vol. 6, no. 4, pp. 467–478, Nov. 2012.

[9] J. Moreira, M. Fernández, and G. Kabatiansky, "Fingerprinting basado en códigos cuasi separables con identificación eficiente," in *Proc. Jornadas de Ingeniería Telemática (JITEL)*, Granada, Spain, Oct. 2013 (To appear).

[10] J. Moreira, M. Fernández, and G. Kabatiansky, "Constructions of almost secure-frameproof codes based on small-bias probability spaces," in *Proc. Int. Workshop Security (IWSEC)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 8231, Okinawa, Japan, Nov. 2013, pp. 53–67.

[11] N. R. Wagner, "Fingerprinting," in *Proc. IEEE Symp. Security, Privacy (SP)*, Oakland, CA, Apr. 1983, pp. 18–22.

[12] G. R. Blakley, C. Meadows, and G. B. Purdy, "Fingerprinting long forgiving messages," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 218, Santa Barbara, CA, Aug. 1985, pp. 180–189.

[13] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 963, Santa Barbara, CA, Aug. 1995, pp. 452–465.

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

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

[16] M. Fernández and M. Soriano, "Fingerprinting concatenated codes with efficient identification," in *Proc. Int. Conf. Inform. Security (ISC)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 2433, Sao Paulo, Brazil, Sep. 2002, pp. 459–470.

[17] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[18] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 1–24, May 2008.

[19] P. Moulin and R. Koetter, "Data-hiding codes," *Proc. IEEE*, vol. 93, no. 12, pp. 2083–2127, Dec. 2005.

[20] A. D. Friedman, R. L. Graham, and J. D. Ullman, "Universal single transition time asynchronous state assignments," *IEEE Trans. Comput.*, vol. C-18, no. 6, pp. 541–547, Jun. 1969.

[21] Y. L. Sagalovich, "Separating systems," *Probl. Inform. Transm.*, vol. 30, no. 2, pp. 105–123, 1994.

[22] M. S. Pinsker and Y. L. Sagalovich, "Lower bound on the cardinality of code of automata's states," *Probl. Inform. Transm.*, vol. 8, no. 3, pp. 59–66, 1972.

[23] Y. L. Sagalovich, "Completely separating systems," *Probl. Inform. Transm.*, vol. 18, no. 2, pp. 140–146, 1982.

[24] J. Körner and G. Simonyi, "Separating partition systems and locally different sequences," *SIAM J. Discr. Math. (SIDMA)*, vol. 1, no. 3, pp. 355–359, Aug. 1988.

[25] G. D. Cohen and H. G. Schaathun, "Asymptotic overview on separating codes," Deptartment of Informatics, University of Bergen, Norway, Tech. Rep. 248, Aug. 2003.

[26] G. D. Cohen and H. G. Schaathun, "Upper bounds on separating codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1291–1294, Jun. 2004.

[27] D. R. Stinson, T. van Trung, and R. Wei, "Secure frameproof codes, key distribution patterns, group testing algorithms and related structures," *J. Stat. Plan. Infer.*, vol. 86, no. 2, pp. 595–617, May 2000.

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

[29] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 839, Santa Barbara, CA, Aug. 1994, pp. 480–491.

[30] B. Chor, A. Fiat, M. Naor, and B. Pinkas, "Tracing traitors," *IEEE Trans. Inf. Theory*, vol. 46, no. 3, pp. 893–910, May 2000.

[31] H. D. L. Hollmann, J. H. van Lint, J.-P. Linnartz, and L. M. G. M. Tolhuizen, "On codes with the identifiable parent property," *J. Combin. Theory, Ser. A*, vol. 82, no. 2, pp. 121–133, May 1998.

[32] N. P. Anthapadmanabhan, A. Barg, and I. Dummer, "On the fingerprinting capacity under the marking assumption," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2678–2689, Jun. 2008.

[33] J. Cotrina-Navau and M. Fernández, "A family of asymptotically good binary fingerprinting codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5335–5343, Oct. 2010.

[34] G. D. Forney, *Concatenated codes*. Cambridge, MA: MIT Press, 1966.

[35] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963.

[36] A. Silverberg, J. Staddon, and J. L. Walker, "Efficient traitor tracing algorithms using list decoding," in *Proc. Int. Conf. Theory, Appl. Cryptol., Inf. Security (ASIACRYPT)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 2248, Gold Coast, Australia, Dec. 2001, pp. 175–192.

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

[38] P. Elias, "List decoding for noisy channels," Res. Lab. Electron., MIT, Cambridge, MA, Tech. Rep. 335, Sep. 1957.

[39] J. M. Wozencraft, "List decoding," Res. Lab. Electron., MIT, Cambridge, MA, Tech. Rep., 1958.

[40] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity*, vol. 13, no. 1, pp. 180–193, Mar. 1997.

[41] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sept. 1999.

[42] V. Guruswami, "List decoding of error-correcting codes," Ph.D. dissertation, Dept. Elect. Eng., Comp. Science, MIT, Cambridge, MA, 2001.

[43] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Sorrento, Italy, Jun. 2000, p. 61.

[44] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[45] M. Fernandez and M. Soriano, "Identification of traitors in algebraic-geometric traceability codes," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 3073–3077, Oct. 2004.

[46] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *SIAM J. Appl. Math. (SIAP)*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[47] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: Elsevier/North-Holland, 1977.

[48] J. van Lint, *Introduction to Coding Theory (Graduate Texts in Mathematics)*, 3rd ed. Berlin, Germany: Springer-Verlag, 1999.

[49] R. Safavi-Naini and Y. Wang, "Collusion secure q-ary fingerprinting for perceptual content," in *Proc. ACM Workshop Digit. Rights Management (DRM)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 2320, Philadelphia, PA, Nov. 2001, pp. 57–75.

[50] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *manuscript*, 2000.

[51] E. Berlekamp, "Bounded distance+1 soft-decision Reed-Solomon decoding," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 704–720, May 1996.

[52] N. P. Anthapadmanabhan, "Random codes and graphs for secure communication," Ph.D. dissertation, Dept. Elect. Comp. Eng., Univ. of Maryland, College Park, MD, 2009.

[53] N. Anthapadmanabhan and A. Barg, "Two-level fingerprinting: Stronger definitions and code constructions," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 2528–2532.

[54] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, 2nd ed. Cambridge, UK: Cambridge Univ. Press, 2011.

[55] G. R. Blakley and G. Kabatiansky, "Random coding technique for digital fingerprinting codes: fighting two pirates revisited," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Chicago, IL, Jun. 2004, p. 203.

[56] J. Bierbrauer and H. Schellwat, "Almost independent and weakly biased arrays: Efficient constructions and cryptologic applications," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 1880, Santa Barbara, CA, Aug. 2000, pp. 533–544.

[57] J. Naor and M. Naor, "Small-bias probability spaces: Efficient constructions and applications," *SIAM J. Comput. (SICOMP)*, vol. 22, no. 4, pp. 838–856, Aug. 1993.

[58] N. Alon, O. Goldreich, J. Håstad, and R. Peralta, "Simple constructions of almost k-wise independent random variables," *Random Struct. Alg.*, vol. 3, no. 3, pp. 289–304, 1992.

[59] U. V. Vazirani, "Randomness, adversaries and computation," Ph.D. dissertation, Dept. Elect. Eng. Comp. Sci., Univ. California, Berkeley, 1986.

[60] P. Diaconis, *Group Representations in Probability and Statistics*. Beachwood, OH: Inst. Math. Stat., 1988.

[61] N. Alon, V. Guruswami, T. Kaufman, and M. Sudan, "Guessing secrets efficiently via list decoding," *ACM Trans. Alg.*, vol. 3, no. 4, pp. 1–16, Nov. 2007.

[62] H. Jin and M. Blaum, "Combinatorial properties for traceability codes using error correcting codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 804–808, Feb. 2007.

[63] M. Fernandez, J. Cotrina, M. Soriano, and N. Domingo, "A note about the identifier parent property in Reed-Solomon codes," *Comput., Security*, vol. 29, no. 5, pp. 628–635, Jul. 2010.

[64] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, revised ed. UK: Cambridge University Press, 1994.