MASTER THESIS

Properties and evaluation of fingerprinting codes

José Moreira Sánchez
Advisor: Marcel Fernández Muñoz

Escola Tècnica Superior d'Enginyeria de
Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
September 2009

Part of this work has been accepted for the First IEEE Workshop on Information Forensics and Security sponsored by the IEEE Signal Processing Society (WIFS09). December 6-9, 2009 - London, United Kingdom [1].

Summary

The concept of data fingerprinting is of paramount importance in the framework of digital content distribution. This project deals with fingerprinting codes, which are used to prevent dishonest users from redistributing copyrighted material. After introducing some basic notions of coding and fingerprinting theory, the project is divided in two parts.

In the first part, we present and analyze some of the main existing fingerprinting codes and we also discuss some new constructions. The study is specifically focused on the estimation of the minimum length of the codes, given the design parameters of the system: number of users to allocate, maximum size of the collusions and probability of identification error. Also, we present some theoretical results about the new code construction studied. Finally, we present several simulations, comparing the different codes and estimating what is the minimum-length code in each region.

The second part of the project is devoted to the study of the properties of Reed-Solomon codes in the context of fingerprinting. Codes with the traceability (TA) property are of remarkable significance, since they provide an efficient way to identify traitors. Codes with the identifiable parent property (IPP) are also capable of identifying traitors, requiring less restrictive conditions than the TA codes at the expense of not having an efficient decoding algorithm, in the general case. Other codes that have been widely studied but possess a weaker traitor-tracing capability are the secure frameproof codes (SFP). It is a well-known result that TA implies IPP and IPP implies SFP. The converse is in general false. However, it has been conjectured that for Reed-Solomon codes all three properties are equivalent. In this paper we investigate this equivalence, and provide a positive answer for families of Reed-Solomon codes when the number of traitors divide the size of the code field.

Keywords: Fingerprinting, Traitor Tracing, Identifiable Parent Property, Secure Frameproof Property, Simplex Codes, Boneh-Shaw Codes, Tardos Codes, Barg et al. Codes, Reed-Solomon Codes, Algebraic-Geometric Codes.

Acknowledgements

I would like to thank specially my thesis advisor, Marcel Fernández for his help and support. Also, I would like to thank my family, friends and workmates in Genaker and ISG-UPC.

Contents

1

Preface

This project deals with the study of the parameters and properties of fingerprinting codes, and it is specifically focused on the length of the codes. There are many families of such codes, and in this study we have considered some of the families which have high relevance in this field. The project is organized as follows.

In section 2 we introduce the topic of fingerprinting and define some useful notation related to coding theory, fingerprinting and traceability codes. We also present some basic and well-known results about fingerprinting codes.

Section 3 is devoted to the analysis of the the proposed binary fingerprinting codes. In this section, we present an analysis of the length of the Boneh-Shaw codes, some results related to secure frameproof codes and we propose methods to determine the length of different versions of the Barg codes. Also, a new family of codes, presented in [2], is studied. We conclude this section with a comparative analysis between the different families of codes.

In section 4 we study the traceability properties of Reed-Solomon codes.In 2001, Staddon et al. [3] raised a question concerning this topic. Essentially, one can classify codes according to their capacity to identify dishonest users. Obviously, codes with "weaker" tracing properties are a subset of codes with "stronger" tracing properties. The question raised in [3] asks wether, for the case of Reed-Solomon codes, all these properties are equivalent. We give a positive answer to this question for a large family of Reed-Solomon codes.

1.1 Notation

Here we present a summary of the notation that we use in the report of the project.

Symbol/acronym	Description
$\mathrm{D}(\sigma\|p)$	Kullback-Leibler divergence of two binomial distributions, $\mathrm{D}(\sigma\|p) = \sigma \log_2(\sigma/p) + (1-\sigma)\log_2((1-\sigma)/(1-p))$.
$\mathbb{F}_q$	The finite field of q elements.
$\mathbb{F}_q^*$	The multiplicative group of $\mathbb{F}_q$.
$\mathbb{F}_q[x]$	The ring of univariate polynomials over $\mathbb{F}_q$.
$\mathbb{F}_q[x]_k$	The ring of univariate polynomials over $\mathbb{F}_q$ of degree $\leq k$.
$\mathbf{a}, \mathbf{b}, \ldots$	Vectors over a finite field (boldface).
$\mathbf{a}^{(i)}$	Cyclic rotation in i coordinates to the right of $\mathbf{a} \in \mathbb{F}_q^n$.
$\mathrm{d}(\mathbf{a}, \mathbf{b})$	Hamming distance between the vectors $\mathbf{a}$ and $\mathbf{b}$.
$\mathrm{w}(\mathbf{a})$	Weight of the codeword $\mathbf{a}$.
$\mathrm{w}_B(\mathbf{a})$	Weight of the codeword $\mathbf{a}$ restricted to the set of coordinates in B.
$\mathrm{D}(A, B)$	Group separation between the sets A and B.
$(n, M, d)_q$-code	A code of length n, size M and minimum distance d.
$[n, k, d]_q$-code	A linear code of length n, dimension k and minimum distance d.
κ	Normalized dimension (rate) of a code: $(\log_q M)/k$.
δ	Normalized minimum distance of a code: n/d.
$\mathcal{C}_o \circ \mathcal{C}_i$	Concatenation of the outer code $\mathcal{C}_o$ with the inner code $\mathcal{C}_i$.
$\mathcal{P}_q(n, k)$	Polynomial code of length n and dimension k over $\mathbb{F}_q$.
$\mathcal{RS}_q(k)$	Reed-Solomon code of dimension k over $\mathbb{F}_q$.
$\mathcal{ERS}_q(k)$	Extended Reed-Solomon code of dimension k over $\mathbb{F}_q$.
$\mathcal{AG}_q(n, k, d)$	Algebraic-geometric code approaching the Tsfasman-Vlăduţ-Zink bound, of length n, dimension k and minimum distance d over $\mathbb{F}_q$.
$\mathcal{S}_q(k)$	Simplex code of dimension k over $\mathbb{F}_q$.
$\mathcal{FS}(n_o, k_i)$	Binary polynomial simplex concatenated code defined in [4] (Fernández-Soriano) with outer code length n_o and inner dimension k_i.
$\mathcal{BS}(M, \epsilon)$	Binary M-secure Boneh-Shaw code with error ϵ, of size M.
$\mathcal{BS}(M, c, \epsilon)$	Binary c-secure Boneh-Shaw code with error ϵ, of size M.
$\mathcal{BS}^\star(M, c, \epsilon)$	Binary c-secure concatenated Boneh-Shaw code with error ϵ.
$\mathcal{B}(M, c, \epsilon)$	Binary c-secure Barg code with error ϵ, of size M.
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	Barg code with outer Reed-Solomon code.

Symbol/acronym	Description
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	Barg code with outer algebraic-geometric code.
$\mathcal{CF}(M, c, \epsilon)$	Binary c-secure code defined in [2] (Cotrina-Fernández) with error ϵ, of size M.
$\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$	Code defined in [2] with outer Reed-Solomon code.
$\mathcal{CF}_{\mathrm{AG}}(M, c, \epsilon)$	Code defined in [2] with outer algebraic-geometric code.
$\mathcal{T}(M, c, \epsilon)$	Binary c-secure Tardos code with error ϵ, of size M.
SFP	Secure frameproof property.
IPP	Identifiable parent property.
TA	Traceability property.
MDS	Maximum distance separable (code).
TVZ	Tsfasman-Vlăduţ-Zink (bound).

2

Preliminaries

Fingerprinting and watermarking are methods to prevent copyright violations and illegal content redistribution, respectively. Watermarking has been an effective tool for centuries. For example, in the French Decorations scandal of 1887, a paper watermark established that two political letters supposedly written in 1884 were actually written on a paper manufactured in 1885.

Digital watermarking is the process of embedding copyright information (watermark) into a digital content which is going to be distributed to a set of users. The content may be audio, pictures or video, for instance. If the content is copied, then the watermark is also carried in the copy. Therefore, the legitimate author has a tool to fight against false claims of authorship. The watermarking process should ideally meet the two following requirements:

- The watermark must be either imperceptible or, if it is not, it must be non-intrusive.
- It must be a robust process, i.e. the watermark must not be easily made unreadable or deleted.

We shall assume unless otherwise stated that we posses an ideal watermarking algorithm.

Fingerprinting is also an old cryptographic technique. Several hundred years ago, distributors of logarithm tables used to introduce tiny errors in the insignificant digits of $\log x$ for few random values of x. Had an owner of a logarithm table sold illegal copies of it, the errors in the table would have allowed to identify who was that owner.

In the framework of digital content distribution, illegal redistribution is a major concern. Therefore, the digital fingerprinting technique appears as a method to discourage it. In this case, the distributor embeds in the digital content, using a watermarking algorithm, an unique piece of information

(fingerprint) for each user. If the content is illegally redistributed, the fingerprint can be extracted and identify the dishonest user. Again, the users may try to damage the fingerprint before they redistribute the content. This, however, should not cause much worry to the distributor if the watermarking process is robust.

Nevertheless, the fingerprinting scenario is prone to another kind of attack known as collusion attack. As the copies of the content owned by the users contain different fingerprints they are, essentially, different objects. Several users (traitors) may compare their copies and find the locations where they differ. This simple operation reveals where part of the marks are located. Traitors may generate a new copy of the content where these locations are deleted or modified in order not to being caught. This has an additional and more severe problem: the pirate copy generated by the traitors may be very similar or coincide with that of an innocent user. Note that the traitors are unable to detect fingerprint positions where their copies agree. The distributor, with that amount of information, would like to be able to identify at least one of the traitors. Therefore, and assuming that we have robust watermark algorithms, we are interested in the design of sets of fingerprints (fingerprinting codes) who are resistant against collusion attacks.

2.1 Coding theory definitions

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

Given a power of a prime number, q, we denote the finite field of q elements by $\mathbb{F}_q$. For any integer $n \geq 1$ we denote the elements of $\mathbb{F}_q^n$ in boldface, e.g. $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$. As we have just shown, a_i represents the ith coordinate of the vector $\mathbf{a}$. For any subset $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots\} \subseteq \mathbb{F}_q^n$, we denote by $\mathrm{u}(A)$ and $\mathrm{m}(A)$ the sets of the unmatching and matching coordinates of the elements of A:

$$\mathrm{u}(A) = \{i : a_{j,i} \neq a_{k,i} \text{ for some } \mathbf{a}_j, \mathbf{a}_k \in A\}$$
$$\mathrm{m}(A) = \{i : a_{j,i} = a_{k,i} \text{ for all } \mathbf{a}_j, \mathbf{a}_k \in A\}.$$

The Hamming distance (or simply, the distance) between $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^n$ is defined as $\mathrm{d}(\mathbf{a}, \mathbf{b}) = |\,\mathrm{u}(\{\mathbf{a}, \mathbf{b}\})|$, and the similitude between $\mathbf{a}$ and $\mathbf{b}$ as $\mathrm{s}(\mathbf{a}, \mathbf{b}) = |\,\mathrm{m}(\{\mathbf{a}, \mathbf{b}\})|$. It is usual to generalize these two concepts for nonempty subsets

of vectors $A, B \subseteq \mathbb{F}_q^n$ as

$$\mathrm{d}(A,B) = \min\{\mathrm{d}(\mathbf{a},\mathbf{b}) : a \in A, b \in B\}$$
$$\mathrm{s}(A,B) = \max\{\mathrm{s}(\mathbf{a},\mathbf{b}) : a \in A, b \in B\}.$$

For the analysis of fingerprinting codes it will also be useful to define the following concept related to the distance between codewords.

Definition 2.1.1. Let A, B be two nonempty subsets of $\mathbb{F}_q^n$, $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots\}$, $B = \{\mathbf{b}_1, \mathbf{b}_2, \dots\}$. We define the group separation between A and B, $\mathrm{D}(A,B)$, as the number of coordinates where the elements of A and B have disjoint elements of $\mathbb{F}_q$, that is

$$\mathrm{D}(A,B) = |\,\mathrm{U}(A,B)|,$$

where $\mathrm{U}(A,B) = \{i : \{a_{1,i}, a_{2,i}, \dots\} \cap \{b_{1,i}, b_{2,i}, \dots\} = \emptyset\}$ is the group unmatching set. The coordinates in $\mathrm{U}(A,B)$ are called separated coordinates, whereas the remaining are called nonseparated coordinates.

One can easily verify that the relations $\mathrm{d}(A,B) + \mathrm{s}(A,B) = n$ and $\mathrm{d}(A,B) \geq \mathrm{D}(A,B)$ always hold. Whenever $|A| = |B| = 1$ the last relation is satisfied with equality.

An $(n, M, d)_q$-block code $\mathcal{C}$ is a nonempty subset of $\mathbb{F}_q^n$ of size M, where $d = \min\{\mathrm{d}(\mathbf{a},\mathbf{b}) : \mathbf{a},\mathbf{b} \in \mathcal{C},\ \mathbf{a} \neq \mathbf{b}\}$ is called the minimum distance of the code. We will only deal with block codes, therefore, the adjective *block* will be omitted henceforth. We refer to the elements of $\mathcal{C}$ as codewords. If $\mathcal{C}$ is a linear k-dimensional vector space over $\mathbb{F}_q$ we say that $\mathcal{C}$ is an $[n,k,d]_q$-code.

Given two integers c_1 and c_2, we denote by D_{c_1,c_2} the smallest of the $\mathrm{D}(A,B)$ between disjoint sets A, B of an $(n, M, d)_q$-code with $|A| = c_1$ and $|B| = c_2$. Clearly, $D_{1,1}$ is the minimum distance of the code, d. The value D_{c_1,c_2} is called the (c_1, c_2)-group separation of the code. For A and B disjoint, we will say that they form a (c_1, c_2)-nonseparated configuration whenever they satisfy $\mathrm{D}(A,B) = 0$.

Finally, we define the concept of code concatenation, which will be exploded throughout the project.

Definition 2.1.2. Given an $(n_o, M, d_o)_q$-code $\mathcal{C}_o$ (called outer code) and an $(n_i, q, d_i)_{q_i}$-code $\mathcal{C}_i$ (called inner code) we denote by $\mathcal{C}_o \circ \mathcal{C}_i$ the $(n_o n_i, M, d_o d_i)_{q_i}$-code $\mathcal{C}$ constructed as

$$\mathcal{C} = \{(\phi_1(u_1) \| \cdots \| \phi_{n_o}(u_{n_o})) : \mathbf{u} = (u_1, \dots, u_{n_o}) \in \mathcal{C}_o\},$$

where ϕ_j is a bijective mapping $\phi_j : \mathbb{F}_q \to \mathcal{C}_i$ and $\|$ denotes the concatenation of codewords. We say that $\mathcal{C}$ is the concatenated code of $\mathcal{C}_o$ and $\mathcal{C}_i$.

In other words, to construct a codeword of $\mathcal{C} = \mathcal{C}_o \circ \mathcal{C}_i$, we choose a codeword of $\mathcal{C}_o$, $\mathbf{u}$, and for every coordinate j we replace u_j by the value $\phi_j(u_j)$. The corresponding codeword of $\mathcal{C}$ is the concatenation of the $\phi_j(u_j)$'s. It is easy to see that $\mathcal{C}$ has length $n_o n_i$, size M and minimum distance $d_o d_i$. If $\mathcal{C}_o$ and $\mathcal{C}_i$ are $[n_o, k_o, d_o]_q$ and $[n_i, k_i, d_i]_{q_i}$-codes respectively, the code $\mathcal{C}$ has dimension $k = k_o k_i$.

2.2 Fingerprinting and traceable codes

Assume that a distributor is applying the fingerprinting technique, i.e. the copies of some content are being watermarked with a fingerprinting code $\mathcal{C}$, and each fingerprint $\mathbf{u}_i \in \mathcal{C}$ is assigned to a user. We will make no distinction between users and their corresponding fingerprints. Considering that a set of c traitors $C = \{\mathbf{t}_1, \ldots, \mathbf{t}_c\}$ collude and construct a pirate object, we are interested in the properties of the pirate fingerprint $\mathbf{x}$ produced by them.

Definition 2.2.1. Given an $(n, M, d)_q$-fingerprinting code $\mathcal{C}$, the envelope of $C \subseteq \mathcal{C}$, denoted by $\mathfrak{E}(C)$, is the set of pirate fingerprints that can be produced by the codewords in C in a collusion attack. We denote by $\mathfrak{E}_c(\mathcal{C})$ the set of all the pirate fingerprints that can be generated by coalitions of size at most c:

$$\mathfrak{E}_c(\mathcal{C}) = \bigcup_{\substack{C \subseteq \mathcal{C} \\ |C| \leq c}} \mathfrak{E}(C).$$

Obviously, each $\mathbf{x} \in \mathfrak{E}(C)$ must be equal to each $\mathbf{t}_i \in C$ in the set of matching coordinates, $\mathrm{m}(C)$. This is known as the marking assumption. For the coordinates in $\mathrm{u}(C)$, several models can be defined:

- The narrow-sense envelope model: the symbol at each position of the pirate fingerprint can only be one of the symbols that the traitors have at that position:

$$\mathfrak{E}(C) = \{\mathbf{x} \in \mathbb{F}_q^n : x_j \in \{t_{1,j}, \ldots, t_{c,j}\}\}.$$

- The wide-sense envelope model: the traitors can put an arbitrary element of $\mathbb{F}_q$ in the coordinates in $\mathrm{u}(C)$:

$$\mathfrak{E}(C) = \{\mathbf{x} \in \mathbb{F}_q^n : x_j = t_{1,j} \text{ for } j \in \mathrm{m}(C)\}.$$

- The expanded narrow-sense envelope model: the traitors can either put one of the symbols that the traitors have at that position or make it

unreadable. This is denoted by $*$, an erasure symbol:

$$\mathfrak{E}(C) = \{\mathbf{x} \in (\mathbb{F}_q \cup \{*\})^n : x_j = t_{1,j} \text{ for } j \in \mathrm{m}(C) \text{ and } x_j \in \{*, t_{1,j}, \ldots, t_{c,j}\} \text{ for } j \in \mathrm{u}(C)\}.$$

- The expanded wide-sense envelope model: the traitors can put an arbitrary element of $\mathbb{F}_q \cup \{*\}$ in the coordinates in $\mathrm{u}(C)$:

$$\mathfrak{E}(C) = \{\mathbf{x} \in (\mathbb{F}_q \cup \{*\})^n : x_j = t_{1,j} \text{ for } j \in \mathrm{m}(C)\}.$$

Definition 2.2.2. Given a fingerprinting code $\mathcal{C}$, a σ-strategy for a set of traitors, $C \subseteq \mathcal{C}$, is a randomized or deterministic algorithm that takes as input C and outputs a pirate codeword $\mathbf{x} = \sigma(C) \in \mathfrak{E}(C)$

The presented scenarios are discussed in more detail in [5]. Intuitively, the wide-sense envelopes lead to more sophisticated σ-strategies than the narrow-sense envelopes. The same is valid for non-expanded versus expanded envelopes. An important remark is that, as we are interested in the case of digital content distribution, we are interested in fingerprinting codes over $\mathbb{F}_2$. In this case the four models are equivalent in terms of traitor tracing. What is more, for binary codes it is detrimental for the traitors to use $*$, since it gives the distributor more information that merely inserting 0 or 1 in a detectable position.

Definition 2.2.3. Consider the narrow-sense envelope model. Then, an $(n, M, d)_q$-fingerprinting code $\mathcal{C}$ may have the following properties:

- The code $\mathcal{C}$ has the (c_1, c_2)-secure frameproof property (SFP) if its group separation satisfies $D_{c_1,c_2} > 0$. In other words, for any $C_1, C_2 \subseteq \mathcal{C}$ with $|C_1| = c_1$ and $C_2 = c_2$ it holds that

$$C_1 \cap C_2 = \emptyset \Rightarrow \mathfrak{E}(C_1) \cap \mathfrak{E}(C_2) = \emptyset.$$

- The code $\mathcal{C}$ has the c-identifiable parent property (IPP) if for any $\mathbf{x} \in \mathbb{F}_q^n$ either $\mathbf{x} \notin \mathfrak{E}_c(\mathcal{C})$ or the intersection of all the coalitions capable of generating $\mathbf{x}$ is not empty,

$$\bigcap_{\substack{C \subseteq \mathcal{C}, |C| \leq c \\ \mathbf{x} \in \mathfrak{E}(C)}} C \neq \emptyset.$$

- The code has the c-traceability (TA) property if for any $C \subseteq \mathcal{C}$ with $|C| = c$ and $\mathbf{x} \in \mathfrak{E}(C)$, there exists some $\mathbf{t} \in C$ such that $\mathrm{d}(\mathbf{x}, \mathbf{t}) < \mathrm{d}(\mathbf{x}, \mathbf{y})$, for any $\mathbf{y} \in \mathcal{C} \setminus C$.

The ideas under the previous definitions are the following. If a code has the $(1, c)$-SFP property,[1] then, no coalition of size at most c will be able to generate the fingerprint of any user. However, they may generate a pirate codeword $\mathbf{x}$ and claim that it was generated by another c-coalition. With an (c, c)-SFP code [3, 6] they would not be able to accuse a completely disjoint coalition. Anyways, this does not guarantee that some traitor may be caught. If the fingerprints belong to a c-IPP code [7, 8] then, one can ensure that, at least, one traitor will be caught: if a codeword belongs to the intersection of all the coalitions that can generate a pirate codeword, in particular, it belongs to the coalition that actually generated it. Regarding c-TA codes [9, 3], they offer the same level of security than c-IPP codes, with the additional benefit that some traitor can be identified efficiently, as it is the closest codeword to $\mathbf{x}$. The following relations are well-known results [3]:

$$c\text{-TA} \Rightarrow c\text{-IPP} \Rightarrow (c, c)\text{-SFP}. \tag{2.1}$$

What is more, for any $(n, M, d)_q$-code

$$d > n(1 - 1/c^2) \Rightarrow c\text{-TA}. \tag{2.2}$$

Definition 2.2.4. We say that a fingerprinting code $\mathcal{C}$ (under any envelope model), with identification algorithm ρ, is c-secure with error ϵ if for any set of traitors $C \subseteq \mathcal{C}$, with $|C| \leq c$, using any σ-strategy the probability that either no traitor is caught or some innocent user is accused is less than ϵ, i.e.

$$P(\rho(\sigma(C)) = \emptyset \vee \rho(\sigma(C)) \cap \mathcal{C} \setminus C \neq \emptyset) < \epsilon.$$

Note that c-IPP and c-TA codes can be viewed as c-secure fingerprinting codes with 0 error.[2] One may think that they represent the solution to the fingerprinting problem. However, these codes have two drawbacks: they are restricted to the narrow-case scenario and the size of the field limits severely their collusion-resistant properties.

Lemma 2.2.5 ([3])**.** For any $(n, M, d)_q$-fingerprinting code with the c-IPP property $c < q$.

As we have commented previously, we are mainly interested in the distribution of digital contents, therefore the codes that we use must be binary. Unfortunately, the previous lemma states that there are not IPP or TA codes (i.e. zero-error codes) over $\mathbb{F}_2$. In the following sections we are devoted to the study of binary fingerprinting codes with error $\epsilon > 0$. We will show how concatenated constructions based on TA codes provide interesting tools for the construction of such codes.

[1] Codes with the $(1, c)$-SFP property are usually called c-frameproof (FP) codes.

[2] Some authors reserve name of *fingerprinting codes* solely for (binary) c-secure fingerprinting codes with error ϵ. They classify IPP and TA codes as *traceable codes*.

3

Analysis of fingerprinting codes

This part of the project is focused on the study of the main binary fingerprinting code families existing in the literature. Our goal is to determine or estimate their length given the following design parameters: the number of users to allocate in the system, M, the maximum size of the collusions, c, and the allowed identification error probability, ϵ. For some of the codes the value of the length follows easily from their definition. For the others, we propose methods to estimate it. Also, an analysis and some results related to a new family of codes are presented. We conclude this part with the simulation results showing the regions where the codes have minimum length.

3.1 SFP and simplex codes

SFP codes have been introduced in section 2.2. They have been studied under the name of separating codes in the context of automata: two systems which transit simultaneously from state a to a' and from b to b' respectively should be forbidden to pass through a common intermediate state. A state is described as an n-bit array, and transiting from the initial state to the final state can only be done through intermediate states by flipping one bit at a time, where the current and the final states differ.

Recall that with a (c_1, c_2)-SFP code $\mathcal{C}$, a size-c_1 coalition $C \subseteq \mathcal{C}$ cannot create a pirate codeword which incriminates a disjoint subset of, at most, c_2 users. We will be mainly interested in SFP codes with $c_1 = c_2 = c$. Whenever $c_1 \neq c_2$ we can obtain a (c, c)-SFP code taking $c = \min\{c_1, c_2\}$, provided that $c \geq 2$.

SFP codes are not very attractive in a fingerprinting scenario. Given a size-c coalition C and a pirate codeword $\mathbf{x} \subseteq \mathfrak{E}(C)$ the only statement that we can make is that the intersection between sets capable of generating $\mathbf{x}$ is

nonempty. In other words if C' is any size-c coalition capable of generating $\mathbf{x}$, all we can say with certainty is that it contains one true traitor, which henceforth implies a high false accusation error probability.

Lemma 3.1.1. A (c, c)-SFP code $\mathcal{C}$ has an identification error probability $\epsilon < 1 - 1/t$.

Proof. Assume that the size-c coalition $C_1 \subseteq \mathcal{C}$ creates the pirate fingerprint $\mathbf{x} = \sigma(C_1) \subseteq \mathfrak{E}(C_1)$. Now, we propose the following tracing algorithm ρ. Consider all the size-c coalitions $C_j \in \mathcal{C}, j = 1, \ldots, m$ such that $\mathbf{x} \in \mathfrak{E}(C_j)$. Choose a codeword $\mathbf{a}$ which belongs to the maximum number of C_j's (ties are broken randomly) and accuse $\mathbf{a}$ as a pirate. As $\mathbf{a}$ belongs to the maximum number of C_j's, say i, and because all the C_j must intersect each other, then, $m \leq i + (t-1)(i-1)$. Therefore,

$$P(\mathbf{a} \in C_1) = \frac{i}{i + (t-1)(i-1)} > \frac{1}{t},$$

which implies $\epsilon < 1 - 1/t$. □

Note that a more accurate approximation of the identification error probability in lemma 3.1.1 can be improved after knowing the maximum value of the intersection in the first stage of the proposed decoding method. In [5] an alternative decoding method is proposed which is somewhat weaker than this result. There, they construct digital fingerprinting codes based on SFP codes, where the concatenation of Reed-Solomon and algebraic-geometric codes strengthen the poor tracing properties of the SFP codes. We will later discuss such codes.

Besides that, SFP codes present additional drawbacks. Little is known about how to construct and decode SFP codes, except maybe for the well-known binary simplex code. What is more, the rate κ of binary SFP codes vanishes dramatically as c increases.

Proposition 3.1.2 ([5])**.** There exist binary (c_1, c_2)-SFP codes of size M and length

$$n \leq \frac{(\log_2 M + 1)(c_1 + c_2 - 1)}{-\log_2(1 - 2^{-c_1 - c_2 + 1})},$$

i.e. of rate

$$\kappa \geq -\frac{\log_2(1 - 2^{-(c_1+c_2-1)})}{c_1 + c_2 - 1} - \frac{1}{M} \simeq -\frac{\log_2(1 - 2^{-(c_1+c_2-1)})}{c_1 + c_2 - 1}. \tag{3.1}$$

If $c_1 = c_2 = c$, for M fixed and c increasing we have that the length of SFP codes increases as $n = \Omega(2^{2c} c \log M)$. The proof of proposition 3.1.2 is based on an exhaustive search of random codes of size n and, therefore, the time complexity for generating such a code is $O(2^{M 2^{2c} c \log M})$. Algorithm 3.1 improves somewhat the running time of the code generation of a binary SFP code. In this case, the algorithm performs a search over all the possible pairs of disjoint subsets, i.e. in

$$\binom{M}{c_1}\binom{M-c_1}{c_2}$$

subsets. For $c_1 = c_2 = c$, the running time of the algorithm is $O(M^{2c})$ at the expense of having a length of order $\Omega(M^{2c})$.

Algorithm 3.1 Simple generation of a binary (c_1, c_2)-SFP code.

Input: Three integers $M \geq 1$, $c_1, c_2 \leq M/2$.
Output: A binary (c, c)-SFP code $\mathcal{C}$ of size M.

```
k ← ⌈log_2 M⌉
C ← C, where C ⊆ F_2^k and |C| = M.
for all A ⊆ C with |A| = c_1 do
  for all B ⊆ C \ A with |B| = c_2 do
    if D(A, B) = 0 then
      for all a ∈ A do
        a ← a‖(1)
      end for
      for all x ∈ C \ A do
        x ← x‖(0)
      end for
    end if
  end for
end for
return C
```

As mentioned before, a notable exception among the SFP codes are the binary simplex codes. They are defined as follows.

Definition 3.1.3. The simplex code of parameter k over $\mathbb{F}_q$, $\mathcal{S}_q(k)$, is the code which has a generator matrix G constructed as the concatenation of $n = (q^k - 1)/(q - 1)$ columns that are pairwise linearly independent vectors of $\mathbb{F}_q^k$.

Example 3.1.4. The simplex code of parameter $k = 3$ over $\mathbb{F}_3$, $\mathcal{S}_3(3)$, is the code generated by

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}.$$

Observe that the n required pairwise linearly independent vectors of $\mathbb{F}_q^k$ can always be found, e.g. by choosing all the non-zero vectors with the leftmost non-zero coordinate equal to 1.

Proposition 3.1.5. The simplex code $\mathcal{S}_q(k)$ satisfies:

- It is an $[n, k, q^{k-1}]_q$-code, with $n = (q^k - 1)/(q - 1)$.
- It meets the Griesmer bound. Therefore it is a linear code with the lowest possible length n for a given dimension k and distance q^{k-1}.
- It is a constant-weight equidistant code.
- It is the dual of a Hamming code.

Now let us focus on the traceability properties of binary simplex codes.

Corollary 3.1.6 ([10])**.** If $\mathcal{C} \subseteq \mathbb{F}_q$ is (c_1, c_2)-SFP, then, $\max\{c_1, c_2\} \leq q$.

Corollary 3.1.7 ([10])**.** All linear, equidistant codes are $(2, 2)$-SFP.

The two previous corollaries imply that the binary simplex codes are $(2, 2)$-SFP, and no better results can be achieved for linear equidistant codes. Simplex codes present several advantages over generic SFP codes: they are linear and they need not be stored, there exist polytime decoding algorithms and, unlike generic $(2, 2)$-SFP codes, they have a remarkable small identification error probability. Any 2 colluding users of a simplex code can not generate a pirate codeword at a distance greater than $d/2$ [11]. Note that this radius is one unit greater than the error-correcting capacity of the code. Therefore, the appropriate decoding algorithm in this case consists of performing a list-decoding in a $d/2 = 2^{r-2}$ radius. Using this decoding algorithm, we have the following result.

Proposition 3.1.8. According to the σ-strategy chosen by a set of traitors, $C = \{\mathbf{t}_1, \mathbf{t}_2\}$ the binary simplex code $\mathcal{S}_2(k)$, with $d = 2^{k-1}$, has the following identification error probability ϵ using a radius-$d/2$ list-decoding algorithm:

- σ_1-strategy: the pirate fingerprint $\mathbf{x}$ is chosen at random from $\mathfrak{E}(C)$, then $\epsilon \leq 2^{k-d}$.

- σ_2-strategy: $\mathbf{x}$ is chosen at random from $\{\mathbf{x}_j : \mathbf{x}_j \in \mathfrak{E}(C) \wedge \mathrm{d}(\mathbf{t}_1, \mathbf{x}) = \mathrm{d}(\mathbf{t}_2, \mathbf{x}) = d/2\}$, then $\epsilon \leq 2^k / \binom{d}{d/2}$.

- σ_3-strategy: $\mathbf{x}$ is chosen at random from $\{\mathbf{x}_j : \mathbf{x}_j \in \mathfrak{E}(C) \wedge \mathrm{d}(\mathbf{t}_1, \mathbf{x}) = \mathrm{d}(\mathbf{t}_2, \mathbf{x}) = \mathrm{w}_U(\mathbf{x}) = d/2\}$, where $U = \mathrm{u}(\mathbf{t}_1, \mathbf{t}_2)$, then $\epsilon \leq 2^k / \binom{d/2}{d/4}$.

The idea of the proof of proposition 3.1.8 is that on a $(2, 2)$-SFP code $\mathcal{C}$ there are only three possible parent configurations for any $\mathbf{x} \in \mathfrak{E}(\mathcal{C})$. Assume that $C_i, i \geq 1$ are subsets of size 2 of $\mathcal{C}$ such that $\mathbf{x} \in \mathfrak{E}(C_i)$. Then, the C_i's must form one of the following configurations:

- Star configuration: all the C_i's intersects in a common element.

- Degenerated star configuration: there exists only one C_i.

- Triangle configuration: there only exist three C_i such that $C_1 = \{\mathbf{a}, \mathbf{b}\}$, $C_2 = \{\mathbf{b}, \mathbf{c}\}$ and $C_3 = \{\mathbf{c}, \mathbf{a}\}$.

One can accuse at least one traitor if the pirate codeword generates a star or a degenerated star configuration. Therefore ϵ is a bound on the generation of a triangle configuration. Unfortunately, given any nonzero codewords of $\mathcal{S}_2(k)$ it is possible to generate deterministically a triangle configuration. This invalidates the use of simplex codes for fingerprinting. However, it can still be used as an inner code in a concatenated construction provided that the resulting code has highly mixed coordinates. A complete discussion of this issues can be found in [11].

As the simplex code is just parameterized just by its dimension, given the design parameters $M, c = 2, \epsilon$, the minimum length of the simplex code, $O(M)$, can be computed as the length of that code with size and identification error probability satisfying the requirements. For $k = 30$ we can allocate $M = 1,07 \cdot 10^9$ users and the greatest identification error probability of proposition 3.1 is $\epsilon \simeq 0$, which are reasonable values.

3.2 Polynomial concatenation of simplex codes

In a paper by M. Fernández and M. Soriano [4] a 2-secure with error ϵ fingerprinting code was presented. It is a binary linear concatenated code based on simplex codes and it is determined by two parameters, n_o and k_i, namely, the size of the outer code and the dimension of the inner code, respectively. First we need the following definition.

Definition 3.2.1. A polynomial code over $\mathbb{F}_q$ of parameters $n \leq q$ and k, $\mathcal{P}_q(n,k)$ is defined as

$$\mathcal{P}_q(n,k) = \{(f(p_1), \ldots, f(p_n) : f \in \mathbb{F}_q[x]_{k-1}\},$$

where $\{p_1, \ldots, p_n\}$ is a size-n set of $\mathbb{F}_q$ and $\mathbb{F}_q[x]_{k-1}$ denotes the ring of the polynomials over $\mathbb{F}_q$ of degree at most $k-1$.

Assume that α is a primitive element of $\mathbb{F}_q$. When $n = q-1$ and $p_i = \alpha^i$ for $i = 1, \ldots, n$ the code is called Reed-Solomon code, and it is denoted by $\mathcal{RS}_q(k)$. If $n = q$ and $p_1 = 0$, $p_i = \alpha^{i-1}$ for $i = 2, \ldots, n$ the code is called extended Reed-Solomon code, and it is denoted by $\mathcal{ERS}_q(k)$.

Proposition 3.2.2. The polynomial code satisfes:

- It is linear, of length n and dimension k.
- It has minimum distance $d = n-k+1$ and, hence, it meets the Singleton bound.
- Reed-Solomon codes are cyclic.

Now, we are in position to define the concatenated construction.

Definition 3.2.3. Given two integer values n_o, k_i with $2^{k_i} \leq n_o$, the concatenated code[1] $\mathcal{FS}(n_o, k_i)$ over $\mathbb{F}_2$ is defined as the concatenation of the polynomial code $\mathcal{P}_{2^{k_i}}(n_o, \lceil n_o/4 \rceil)$ and the binary simplex code $\mathcal{S}_2(k_i)$,

$$\mathcal{FS}(n_o, k_i) = \mathcal{P}(n_o, \lceil n_o/4 \rceil)_{2^{k_i}} \circ \mathcal{S}_2(k_i).$$

To determine the identification error probability, we need to present before the decoding algorithm. It is as follows. The outer code used here is 2-TA, because it satisfies (2.2), and therefore the tracing capacity is limited by the inner code. After decoding the jth subcodeword, we will have a set of 1, 2 or 3 codewords[2] of the simplex code. As a result, and after applying the inverse mapping ϕ_j^{-1}, we will obtain a subset $S_j \subset \mathbb{F}_{2^{k_i}}$ containing up to 3 elements. For $p = 1, 2, 3$, we denote by $S^{(p)}$ the set of the S_j's with $|S_j| = p$. Obviously $|S^{(1)}| + |S^{(2)}| + |S^{(3)}| = n_o$. Next, we construct the following $2^{k_i} \times n_o$ reliability matrix $\Pi = (\pi_{i,j})$, where

$$\pi_{i,j} = \begin{cases} 1/|S_j| & \text{if } \alpha_i \in S_j \\ 0 & \text{otherwise.} \end{cases}$$

[1]Note that when $\mathcal{P}(n_o, \lceil n_o/4 \rceil)$ is a Reed-Solomon code, then $\mathcal{FS}(n_o, k_i)$ is determined by a single parameter.

[2]One codeword if the coalition produced a star configuration, two if it produced a degenerated star and three if it produced a triangle configuration in that subcodword.

Here α_i denotes the ith element of the outer field, according to an arbitrary preestablished order. The matrix Π is used as the input for the Koetter-Vardy soft-decision decoding algorithm [12]. This decoding algorithm returns all the codewords $\mathbf{u} \in \mathcal{P}(n_o, \lceil n_o/4 \rceil)$ that satisfy

$$\frac{\langle \Pi, [\mathbf{u}] \rangle}{\sqrt{\langle \Pi, \Pi \rangle}} \geq \sqrt{k_o - 1} + o(1),$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. For the complete details of the relations between the Koetter-Vardy decoding algorithm and the tracing algorithm of $\mathcal{FS}(n_o, k_i)$, see [4, 11].

Because the outer code is 2-TA, observe that:

1. If $|S^{(1)}|+|S^{(2)}| > 4(n_o-d_o)$, then, at least one of the traitors is identified with probability 1.
2. If $|S^{(2)}| > 2(n_o - d_o)$, then, both traitors are identified with probability 1.
3. If $|S^{(1)}| + |S^{(2)}| <= 4(n_o - d_o)$, then, the only cases of identification are:
   (a) if there exists a j such that there is a $S_j = \{\alpha_1, \alpha_2\} \in S^{(2)}$ and there exist exactly 2 codewords $\mathbf{u}_1, \mathbf{u}_2 \in U$ such that $u_{1,j} = \alpha_1$ and $u_{2,j} = \alpha_2$, output $\mathbf{u}_1, \mathbf{u}_2$ as traitors,
   (b) if there exists a j such that there is a $S_j = \{\alpha\} \in S^{(1)}$ and there exists exactly 1 codeword $\mathbf{u} \in U$ such that $u_j = \alpha$, output $\mathbf{u}$ as a traitor.

Therefore the tracing capacity of the algorithm is limited by the capacity of the traitors to generate triangle configurations in the subcodewords of $\mathcal{FS}(n_o, k_i)$. Taking this into account, the tracing algorithm can only fail if $|S^{(3)}| \geq n_o - 2(n_o - d_o)$.

Given the identification error probability of the inner code, ϵ_i, we define P as

$$P = P(|S^{(3)}| \geq n_o - 2(n_o - d_o)) = \sum_{j=n_o-2(n_o-d_o)}^{n_o} \binom{n_o}{j} \epsilon_i^j (1-\epsilon_i)^{n_o-j}.$$

No codeword will be identified if there is a codeword in the outer code that matches all the parent positions in $S^{(1)} \cup S^{(2)}$. The outer code is a polynomial code over $\mathbb{F}_{2^{k_i}}$, so there are $2^{k_i}\binom{n_o}{k_o-1}$ of such codewords. Since $2^{k_i}\binom{n_o}{k_o-1} \leq 2^{k_i k_o}$, then

$$\epsilon \leq 2^{k_i k_o} \cdot P. \tag{3.2}$$

The original algorithm, however, can be somewhat improved. Assume that U is the set of codewords returned by the Koetter-Vardy algorithm. The idea is that if we can ensure that $\{\mathbf{a}, \mathbf{b}\} \subseteq U$ and we apply a process giving as a result a subset $U' \subseteq U$ ensuring that no traitor will be removed, then if $|U'| = 2$, we can output U' as a positive parent set. An important remark is that the position of the coordinates of every codeword in $\mathcal{FS}(n_o, k_i)$ need to be shuffled (and unshuffled before decoding) to prevent systematic attacks on the simplex code. For a complete discussion of the code, see [4, 11].

Since $\mathcal{FS}(n_o, k_i)$ depends on two parameters, given the design parameters $M, c = 2, \epsilon$, one can perform a search over some pairs of parameters and use the shortest code which satisfies them considering $M \leq 2^{k_i \lceil n_o/4 \rceil}$ and (3.2). For $k_o = 6$ and $n_o = 64$ we obtain a code of size $M = 1,0995 \cdot 10^{12}$, length $n = 992$ and identification error probability $\epsilon \leq 1,0421 \cdot 10^{-66}$, so it is expected that one can find the required code for reasonable values of the design parameters.

3.3 The Boneh-Shaw codes

In [13] a family of binary M-secure with error ϵ codes were introduced by D. Boneh and J. Shaw. They construct the $\mathcal{BS}(M, d)$ for M users using a matrix that has $M - 1$ column types repeated d times each. The codewords are then the rows the matrix.

Example 3.3.1. The $\mathcal{BS}(M, d)$ for $M = 8$ users is

$$
\begin{array}{llccccccc}
 & & \overbrace{}^{B_1} & \overbrace{}^{B_2} & \overbrace{}^{B_3} & \overbrace{}^{B_4} & \overbrace{}^{B_5} & \overbrace{}^{B_6} & \overbrace{}^{B_7} \\
\text{User 1} & \mathbf{u}_1 = (1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1) \\
\text{User 2} & \mathbf{u}_2 = (0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1) \\
\text{User 3} & \mathbf{u}_3 = (0\cdots0 & 0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1) \\
\text{User 4} & \mathbf{u}_4 = (0\cdots0 & 0\cdots0 & 0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1 & 1\cdots1) \\
\text{User 5} & \mathbf{u}_5 = (0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 1\cdots1 & 1\cdots1 & 1\cdots1) \\
\text{User 6} & \mathbf{u}_6 = (0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 1\cdots1 & 1\cdots1) \\
\text{User 7} & \mathbf{u}_7 = (0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 1\cdots1) \\
\text{User 8} & \mathbf{u}_8 = (0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0 & 0\cdots0)
\end{array}
$$

Note that every set of d coordinates, $B_1, \ldots B_{M-1}$, contains only one column type repeated d times.

Proposition 3.3.2 ([13])**.** The $\mathcal{BS}(M, d)$ code satisfies that

- It is a binary (n, M, d)-code with $n = (M - 1)d$.
- If $d \geq 2M^2 \log(2M/\epsilon)$ it is an M-secure fingerprinting code with ϵ error.

The security of the code lies on the uniqueness of the boundaries between 0's and 1's for every user. If user i is not guilty, then, even though the rest of the users collude to generate a pirate codeword $\mathbf{x}$ they will not be able to distinguish between column types i and $i-1$. Therefore, $\mathbf{x}$ will have approximately the same distribution of symbols in these the $B_{i-1} \cup B_i$. This requires, however, that the columns of the codebook are heavily mixed before insert the fingerprints in the content. Algorithm 3.2 describes the original tracing algorithm proposed in [13].

Algorithm 3.2 Tracing algorithm for the $\mathcal{BS}(M, d)$ code.

Input: The $\mathcal{BS}(M, d)$ code with $d \geq 2M^2 \log(2M/\epsilon)$ and a pirate codeword $\mathbf{x}$ generated by any coalition of traitors.

Output: A list of traitors capable of generating the codeword $\mathbf{x}$ with identification error probability ϵ, according to proposition 3.3.2.

$C \leftarrow \emptyset$
if $\mathrm{w}_{B_1}(\mathbf{x}) > 0$ **then**
  $C \leftarrow C \cup \{\mathbf{u}_1\}$
end if
if $\mathrm{w}_{B_{M-1}}(\mathbf{x}) < d$ **then**
  $C \leftarrow C \cup \{\mathbf{u}_M\}$
end if
for all $i = 2, \ldots, M-1$ **do**
  $R_i = B_{i-1} \cup B_i$
  $k \leftarrow \mathrm{w}_{R_i}(\mathbf{x})$
  if $\mathrm{w}_{R_i}(\mathbf{x}) > \frac{k}{2} - \sqrt{\frac{k}{2} \log \frac{2M}{\epsilon}}$ **then**
    $C \leftarrow C \cup \{\mathbf{u}_i\}$
  end if
end for
return C

However, the $\mathcal{BS}(M, d)$ code has the drawback that its length grows as $O(M^3 \log(M/\epsilon)))$, which is not desirable. The cubic growth is due to the fact that the code is designed to fight against maximal coalition sizes. To overcome this problem, under the relations pointed in [14], the following construction is proposed to achieve c-secure codes of logarithmic length, $n = c^{O(1)} \log M$. The idea is to concatenate codes, using as the inner code $\mathcal{C}_i$ the M-secure code $\mathcal{BS}(M_i, d_i)$ and as the outer code $\mathcal{C}_o$ a random (n_o, M_o, d_o)-code over an alphabet of size M_i. The resulting code $\mathcal{BS}^\star(n_o, M_o, M_i, d_i)$ has size $M = M_o$ and length $n = n_o d_i (M_i - 1)$.

Theorem 3.3.3 ([13])**.** Given integers M_o, c and $\epsilon > 0$, set the following values:

- $M_i = 2c$,
- $n_o = 2c \log(2M_o/\epsilon)$,
- $d_i = 2M_i^2 \log(2M_i n_o/\epsilon)$.

Then, the code $\mathcal{BS}^\star(n_o, M_o, M_i, d_i)$ is a c-secure fingerprinting code with ϵ, size $M = M_o$ and length $n = O(n_o d_i M_i)$,

$$n = O(c^4 \log(M/\epsilon) \log(1/\epsilon). \tag{3.3}$$

An important issue of the $\mathcal{BS}^\star(n_o, M_o, M_i, d_i)$ code is that for reasonable values of ϵ the number of codewords is asymptotically $\exp(O(\sqrt{n}))$. This can be seen by taking (3.3) and solving for $\log \epsilon$. We have that

$$\log \epsilon = \frac{c^4 \log M \pm \sqrt{c^8 \log^2 M + 4c^4 n}}{2c^4}.$$

As $\log \epsilon < 0$, we discard the positive root. Now, taking $\log M = O(n^\alpha)$ and substituting it into the previous equation we have that for every value of c

$$\log \epsilon = O(n^\alpha - \sqrt{n + n^{2\alpha}}).$$

If we take $\alpha < 1/2$, then,

$$\log \epsilon = O(-\sqrt{n}). \tag{3.4}$$

For $\alpha \geq 1/2$ we rewrite (3.3) and, because $\log \epsilon \ll n^\alpha$, we obtain

$$n = O(c^4(\log \epsilon - n^\alpha) \log \epsilon) = O(-n^\alpha \log \epsilon).$$

This implies that $\log \epsilon = \Omega(-n^{1-\alpha})$. Combining this with (3.4) we have that

$$\log \epsilon = -\Omega(\min\{\sqrt{n}, n^{1-\alpha}\}).$$

In other words, ϵ cannot decrease faster than $\exp(-\Omega(\sqrt{n}))$, so we take $\alpha = 1/2$ and henceforth $M = \exp(O(\sqrt{n}))$.

Another alternative based on the Boneh-Shaw codes focused in reducing the size of the fingerprinting code is that presented in [15]. There, a new analysis of the $\mathcal{BS}(M, d)$ code is presented, which leads to the following result.

Algorithm 3.3 Tracing algorithm for the modified $\mathcal{BS}(M, d)$ code.

Input: The $\mathcal{BS}(M, d)$ code with $d \geq 8(c + \sqrt{c+1})^2 \log(4M/\epsilon)$ and a pirate codeword $\mathbf{x}$ generated by any coalition of traitors of size at most c.

Output: A list of traitors capable of generating the codeword $\mathbf{x}$ with identification error probability ϵ, according to theorem 3.3.4.

$C \leftarrow \emptyset$
if $\mathrm{w}_{B_1}(\mathbf{x}) > 0$ **then**
  $C \leftarrow C \cup \{\mathbf{u}_1\}$
end if
if $\mathrm{w}_{B_{M-1}}(\mathbf{x}) < d$ **then**
  $C \leftarrow C \cup \{\mathbf{u}_M\}$
end if
$\lambda \leftarrow \sqrt{2d \log\left(\frac{4M}{\epsilon}\right)}$
for all $i = 2, \ldots, M-1$ **do**
  if $\mathrm{w}_{R_i}(\mathbf{x}) - \mathrm{w}_{R_{i-1}}(\mathbf{x}) > 2\lambda$ **then**
    $C \leftarrow C \cup \{\mathbf{u}_i\}$
  end if
end for
return C

Theorem 3.3.4 ([15])**.** The $\mathcal{BS}(M, d)$ code with

$$d \geq 8(c + \sqrt{c+1})^2 \log(4M/\epsilon)$$

that is, of length $O(Mc^2 \log(M/\epsilon))$, is a binary c-secure fingerprinting code with error ϵ.

Applying similar decoding rules as the ones used in [13], Algorithm 3.3 performs the decoding of the redesigned code.

Given the design parameters M, c and ϵ, we denote by $\mathcal{BS}(M, \epsilon)$, $\mathcal{BS}^*(M, c, \epsilon)$, $\mathcal{BS}(M, c, \epsilon)$ the c-secure with error ϵ version of the Boneh-Shaw codes presented in this section, with lengths described in proposition 3.3.2, theorem 3.3.3 and theorem 3.3.4, respectively. Note that for the first code, $c = M$.

3.4 The Barg codes

In [5] A. Barg, G. R. Blakley and G. A. Kabatiansky present a family of digital fingerprinting codes based on (c, c)-SFP codes. As well as in the case of the Boneh-Shaw codes, the idea underneath the Barg codes relies on concatenation.

Definition 3.4.1. Let $\mathcal{C}_o$ be linear $[n_o, k_o, d_o = \delta_o n_o]_q$-code with

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

and $\mathcal{C}_i$ be a size-q code with the (c, c)-SFP property. The Barg code $\mathcal{B}(q, n_o, k_o)$ is defined as the concatenated code $\mathcal{C}_o \circ \mathcal{C}_i$.

Theorem 3.4.2 ([5])**.** The code $\mathcal{B}(q, n_o, k_o)$ using $\mathcal{C}_i$ as the inner code and $\mathcal{C}_o$ as the outer code is a c-secure fingerprinting code of length $n = n_o n_i$ with $M = q^{k_o}$ codewords and identification error probability

$$\epsilon \leq 2^{-n\kappa_i((\log_2 q)^{-1}\, \mathrm{D}(\sigma \| \frac{c-1}{q-1}) - \kappa_o)}. \tag{3.5}$$

Here $\sigma = 1/c - (1-\delta_o)c$ and $\kappa_o = k_o/n_o$, $\kappa_i = k_i/n_i$ are the rate of the outer and the inner code, respectively and $\mathrm{D}(\sigma\|p)$ is the Kullback-Leibler divergence, $\mathrm{D}(\sigma\|\epsilon) = \sigma \log_2(\sigma/\epsilon) + (1-\sigma)\log_2((1-\sigma)/(1-\epsilon))$.

In order to guarantee security against size-c coalitions, the mappings used in the concatenation, $\phi_i : \mathbb{F}_q \to \mathcal{C}_i$, for $i = 1, \ldots, n_o$, must be chosen at random. We now show the two purposed implementations and how their length can be computed.

3.4.1 Reed-Solomon codes as outer codes

The first construction is based on extended Reed-Solomon codes over a large alphabet. Take as inner code $\mathcal{C}_i$ a (c, c)-SFP code of size q and rate κ_i. Next, choose an extended Reed-Solomon code $\mathcal{ERS}_q(k_o)$ as outer code $\mathcal{C}_o$. Now note that, for sufficiently large q, the following approximation can be made:

$$\mathrm{D}\left(\sigma \middle\| \frac{c-1}{q-1}\right) = \sigma \log_2 \left(\frac{\sigma(q-1)}{t-1}\right) + (1-\sigma)\log_2 \left(\frac{1-\sigma}{\left(1 - \frac{t-1}{q-1}\right)}\right) \approx$$

$$\sigma \log_2 \left(\frac{\sigma(q-1)}{t-1}\right) = \sigma(\log_2 \sigma + \log_2(q-1) + \log_2(t-1)) \approx \sigma \log_2 q. \tag{3.6}$$

As $\mathcal{ERS}_q(k_o)$ meets the Singleton bound, and because q is large, its rate and normalized minimum distance satisfy

$$1 - \delta_o = \kappa_o + o(1).$$

This, together with (3.6), implies that the identification error probability of (3.5) can be approximated by

$$\epsilon \leq 2^{-n(c^{-1}\kappa_i - (c+1)\kappa + o(1))}, \tag{3.7}$$

where $\kappa = \kappa_i \kappa_o$ denotes the rate of the concatenated code. The code will exist provided that the exponent in the previous equation is negative, that is, the rate of the outer code must satisfy $\kappa_o < 1/c(c+1)$, and the total rate

$$\kappa < \frac{\kappa_i}{c(c+1)}. \tag{3.8}$$

Now, let us estimate the code length of the previous construction given the design parameters M, c, ϵ. First, note that the rate of the inner code κ_i can be computed according to (3.1), hence, it does not depend on M nor ϵ. Ignoring the term $o(1)$[3] in (3.7) and combining (3.8) with $M = 2^{n\kappa}$ we obtain

$$\frac{\log_2 M}{n} \leq \kappa < \frac{\kappa_i}{c(c+1)}.$$

From (3.7) we obtain

$$n \geq \frac{-\log_2 \epsilon}{\kappa_i/c - (c+1)\kappa},$$

and therefore an estimation of the minimum length of the code $\mathcal{ERS}_q(k_i) \circ \mathcal{C}_i$ for SFP inner codes, according to (3.1), is

$$n \geq 2c(c-1)\frac{-\log_2 \epsilon + (c+1)\log_2 M}{-\log_2(1 - 2^{-(2c-1)})}. \tag{3.9}$$

That is, the code has a length of order $\Omega(\max\{2^{2c}c^2 \log \frac{1}{\epsilon}, 2^{2c}c^3 \log M\})$.

3.4.2 Algebraic-geometric codes as outer codes

The second construction is based on outer algebraic-geometric codes approaching the Tsfasman-Vlăduţ-Zink (TVZ) bound. It is well-known [16] the existence of families of $[n_o, k_o, d_o]_q$ algebraic-geometric codes over a finite field $\mathbb{F}_q$ whose parameters asymptotically approach the bound

$$k_o + d_o \geq n_o - n_o/(\sqrt{q} - 1), \tag{3.10}$$

which can be restated as $\kappa_o + \delta_o = 1 - \frac{1}{\sqrt{q}-1}$. As a consequence, σ is of the form

$$\sigma = \frac{1}{c} - c\left(\kappa_o - \frac{1}{\sqrt{q}-1}\right).$$

Note that the code will exist if

$$\frac{1}{c} - c\left(\kappa_o - \frac{1}{\sqrt{q}-1}\right) > \frac{t-1}{q-1}.$$

[3] This term is actually $1/q$.

This condition can be rewritten as $\kappa_o < A$, where

$$A = \frac{1}{c} - \frac{c-1}{c(q-1)} + \frac{1}{\sqrt{q}-1}. \tag{3.11}$$

The value A is always positive. This is implied by the fact that $c-1+c^2\sqrt{q}+q > 0$, which is a trivial condition. Another necessary condition to guarantee the existence of the code, according to (3.5), is $f(\kappa_o) > 0$, where

$$f(x) = (\log_2 q)^{-1} \, \mathrm{D}\left(\frac{1}{c} - c\left(x - \frac{1}{\sqrt{q}-1}\right) \middle\| \frac{c-1}{q-1}\right) - x.$$

It is easy to see that $f(x)$ is a monotonically decreasing function of x with a root, namely $\kappa_o^{\max}$, in the interval $0 < x < A$. The code will exists for any $\kappa_o < \kappa_o^{\max}$. We omit here the proof because a similar proof will be presented later.

Now, let us compute the length of the Barg code when using an outer algebraic-geometric code. Given the same design parameters, M, c and ϵ, and a set of possible values for q, we proceed as follows.

1. Compute κ_i, which only depends on c.

2. For a given value of q:

   (a) Compute the value $A(q)$ according to (3.11).

   (b) Find the value $\kappa_o^{\max}$ performing a search in the interval $0 < \kappa_o^{\max} < A(q)$. This can be done numerically.

   (c) For every $0 < \kappa_o < \kappa_o^{\max}$ compute $n(\kappa_o, q) = \max\{n_1, n_2\}$, where

   $$n_1 = \log_2(M)/(\kappa_i\kappa_o)$$
   $$n_2 = -\log_2(\epsilon)/(\kappa_i((\log_2 q)^{-1} \, \mathrm{D}(\sigma\|\tfrac{c-1}{q-1}) - \kappa_o)).$$

   This ensures that the code meets the requirements. Compute the value $n(q) = \min\{n(\kappa_o, q) : 0 < \kappa_o < A(q)\}$.

3. Repeat the procedure until all the plausible values of q have been tested and return n, the minimum value of the $n(q)$'s, as the code length.

Note that, the value $n(\kappa_o, q)$ is the minimum length such that both the code size and the identification error probability satisfy the requirements. Therefore, the procedure consists in finding the minimum value of the $n(\kappa_o, q)$'s.

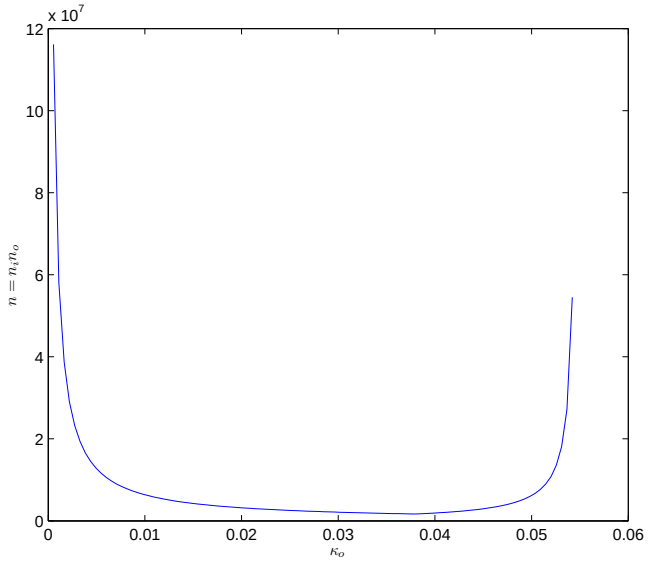


Figure 3.1: Plot of the length of $\mathcal{B}_{\mathrm{AG}}(M, c, \epsilon, q)$ versus κ_o

As an example, in figure 3.1 there is a plot of $n(\kappa_o, q)$ for $M = 10^6, c = 5, \epsilon = 10^{-15}$ and $q = 500$. The minimum length occurs approximately at 0.038 and the minimum achieved code length is $n = 1.6834 \cdot 10^6$. We denote by $\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$ and $\mathcal{B}_{\mathrm{AG}}(M, c, \epsilon)$ the c-secure with error ϵ Barg codes of size M, for outer extended Reed-Solomon and algebraic-geometric codes. The decoding of the Barg codes is performed in a two step-process: first, decoding the inner codes and then, using the list-decoding Guruswami-Sudan [17] decoding algorithm. The algorithm based in a similar idea to that presented for the polynomial concatenated simplex codes.

3.5 The Tardos codes

The original Tardos code was presented in [18]. Currently it is the code with the best-known asymptotic length. In fact, the Tardos code mets the following bound.

Proposition 3.5.1 ([18])**.** Any (n, M, d) c-secure code with error ϵ satisfies

$$n = \Omega(c^2 \log(1/\epsilon)).$$

Definition 3.5.2. Let M, c be positive integers with $c \leq M$ and $0 < \epsilon < 1$. The Tardos code of parameters M, c, ϵ, $\mathcal{T}(M, c, \epsilon)$, is a c-secure fingerprinting

code with error ϵ and length $n = 100c^2\lceil\log(M/\epsilon)\rceil$, where every codeword $\mathbf{u} = (u_1, \ldots, u_n) \in \mathcal{T}(M, c, \epsilon)$ is such that $P(u_i = 1) = p_i$, and every $p_i, 1 \leq i \leq n$, is independently distributed according to the following probability density function:

$$f_{p_i}(p) = \frac{1}{\pi - 4\arcsin(\sqrt{c})} \frac{1}{\sqrt{p(1-p)}}, \qquad \text{with } t = \frac{1}{300c}.$$

In the original paper, it is set a value of $L = 100$ as a multiplicative constant term in the length of the code. However, many researchers found this constant not accurate. Better known approximations are, for instance, $L = 4\pi^2$ [19], $L = 38$ [20]. Other works propose more practical implementations of the Tardos code [21].

Because of its random nature, the decoding of the Tardos code is almost a "brute force" process, i.e. the code must perform a search over all the codewords in $\mathcal{T}(M, c, \epsilon)$. Algorithm 3.4 is the decoding algorithm for this code.

Algorithm 3.4 Tracing algorithm for the $\mathcal{T}(M, c, \epsilon)$ code.

Input: The $\mathcal{T}(M, c, \epsilon)$ code and a pirate codeword $\mathbf{x}$ generated by any coalition of traitors.

Output: A list of traitors capable of generating the codeword $\mathbf{x}$ with identification error probability ϵ.

Arrange the M codewords $\mathbf{u}_1, \ldots, \mathbf{u}_M$ in an $M \times n$ matrix where the jth row is the codeword $\mathbf{u}_j$.

Compute the $M \times n$ matrix U as

$$U_{j,i} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}} & \text{if } u_{j,i} = 1 \\ -\sqrt{\frac{p_i}{1-p_i}} & \text{if } u_{j,i} = 0. \end{cases}$$

$Z \leftarrow 20c\lceil\log(M/\epsilon)\rceil$
$C \leftarrow \emptyset$
for all $\mathbf{u}_i \in \mathcal{T}(M, c, \epsilon)$ **do**
  if $\sum_{i=1}^{n} x_i U_{j,i} > Z$ **then**
    $C \leftarrow C \cup \{\mathbf{u}_j\}$.
  end if
end for
return C

3.6 Structured concatenation of fingerprinting codes

We have already discussed in previous sections constructions based on concatenation. Here we discuss a new family of codes proposed by J. Cotrina and M. Fernández in [2].

As mentioned earlier, TA codes and c-IPP codes can be viewed (under the narrow-case model) as c-secure fingerprinting codes with null error. What is more, TA codes posses and efficient decoding algorithm. Recall from (2.2) that a code with

$$d > n(1 - 1/c^2) \tag{3.12}$$

is c-TA (and hence, c-IPP). Therefore, one may try to find a binary code satisfying (2.2). Unfortunately, for the binary case, we have that not only such codes do not exist, but no binary IPP code do exists.

Corollary 3.6.1. Any $(n, M, d)_q$-code $\mathcal{C}$ over $\mathbb{F}_q$ with $d > n(1-1/c^2)$ satisfies that $c < q$.

Proof. The proof is immediate, since $d > n(1 - 1/c^2) \Rightarrow c$-TA $\Rightarrow c$-IPP $\Rightarrow$ $c < q$, where the last assertion is due to lemma 2.2.5. □

As commented earlier, the field size limits considerably the traceability properties in the narrow-case scenario. Therefore concatenated outer TA codes with inner binary codes cannot be used for our purposes. The idea proposed in [2] consists of concatenating an outer c-TA code $\mathcal{C}_o$ satisfying (3.12) with an inner binary c-secure fingerprinting code with error ϵ. Obviously, after decoding the inner code, we will have an average of $n_o\epsilon_i$ errors in the outer code, and the selected value of d_o may not be higher enough. The solution to this problem consists of adding an additional term that depends on ϵ_i to the value of d_o in (3.12).

Theorem 3.6.2 ([2])**.** Let $\mathcal{C}_o$ be an $(n_o, M, d_o)_q$-code. Let $\mathcal{C}_i$ be an (n_i, q, d_i)-binary c-secure fingerprinting code with ϵ_i error. Then, for any $\epsilon_i < \sigma < 1/(c+1)$, the concatenated code $\mathcal{C} = \mathcal{C}_o \circ \mathcal{C}_i$ is a binary c-secure fingerprinting code with exponentially decreasing identification error probability $\epsilon = \exp(-\Omega(n_o))$ for M users if

$$d_o > n_o \left(1 - \frac{1}{c^2} + \frac{\sigma(c+1)}{c^2}\right). \tag{3.13}$$

The proof of the theorem, as well as in the case of the Barg codes, is based on the well-known Chernoff bound:

$$P\left(\sum_{i=1}^{n_o}\xi_i \geq n_o\sigma\right) \leq 2^{-n_o\,\mathrm{D}(\sigma\|\epsilon_i)},$$

where ξ_i are independent Bernoully random variables equal to 1 with probability ϵ_i and 0 with probability $1-\epsilon_i$. Using this result, the identification error probability of the concatenated code can be bounded by

$$\epsilon \leq 2^{-n_o\,\mathrm{D}(\sigma\|\epsilon_i)} \tag{3.14}$$

In order to obtain a short fingerprinting code and to provide an efficient decoding algorithm, the two purposed outer codes are based, as in the case of the Barg codes, in Reed-Solomon codes and algebraic-geometric codes approaching the TVZ bound. As inner codes, Boneh-Shaw codes $\mathcal{BS}(q,c,\epsilon)$ are proposed. Given the design parameters, we denote by $\mathcal{CF}_{\mathrm{RS}}(M,c,\epsilon)$ and $\mathcal{CF}_{\mathrm{AG}}(M,c,\epsilon)$ the concatenated codes, when using Reed-Solomon and algebraic-geometric as outer codes, respectively.

3.6.1 Reed-Solomon codes as outer codes

Our goal is to choose the optimal parameters to construct a minimum-length concatenated code $\mathcal{CF}_{\mathrm{RS}}(M,c,\epsilon) = \mathcal{RS}_q(k_o)\circ\mathcal{C}_i$, given a fixed family of inner codes $\mathcal{C}_i$. We choose as $\mathcal{C}_o$ a Reed-Solomon code over $\mathbb{F}_q$, where q is the size of the inner code.

Definition 3.6.3. For a fixed value of $0 < \sigma < 1/(c+1)$, we denote by $\mathcal{CF}_{\mathrm{RS}}(M,c,\epsilon,\sigma)$ the c-secure fingerprinting code with error ϵ for M users constructed according to the following steps:

1. Find the minimum prime power q such that $f(q) > 0$, where
$$f(q) = M - q^{\left\lceil (q-1)\left(\frac{1-\sigma(c+1)}{c^2}\right)\right\rceil}.$$
2. Find the $0 < \epsilon_i < \sigma$ such that $g(\epsilon_i) = 0$, where
$$g(\epsilon_i) = q\,\mathrm{D}(\sigma\|\epsilon_i) - \log_2\epsilon.$$
3. Construct $\mathcal{CF}_{\mathrm{RS}}(M,c,\epsilon,\sigma)$ as $\mathcal{RS}_q(k_o)\circ\mathcal{C}_i$, where $\mathcal{RS}_q(k_o)$ is a Reed-Solomon code over $\mathbb{F}_q$ with
$$k_o = \left\lceil n_o\left(\frac{1-\sigma(c+1)}{c^2}\right)\right\rceil,$$
and $\mathcal{C}_i$ is a binary c-secure fingerprinting code with error ϵ_i and size q.

Lemma 3.6.4. Assume that the length of the inner code of $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon, \sigma)$, $\mathcal{C}_i$, is a monotonically decreasing function of ϵ_i. Then, $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon, \sigma)$ exists and it is the shortest c-secure fingerprinting code with error ϵ of size M among all the codes of the form $\mathcal{RS}_q(k_o) \circ \mathcal{C}_i$ for a fixed σ and a fixed family of codes $\mathcal{C}_i$.

Proof. First, note that $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon, \sigma)$ has size, at least, M if the $\mathcal{RS}_q(k_o)$ code has $q^{k_o} \geq M$ codewords. In order to satisfy (3.13) we take

$$d_o = n_o - \left\lceil n_o \left(\frac{1 - \sigma(c+1)}{c^2} \right) \right\rceil + 1,$$

which is the minimum value allowed for d_o, leading to the maximum dimension. Therefore, $M - q^{k_o} > 0$ is precisely the condition that $f(q)$ must satisfy in the first step. As $f(q)$ is a monotonically increasing function of q we can find the smallest value for which $f(q) > 0$. Next, the code must satisfy (3.14). For σ fixed, $\mathrm{D}(\sigma \| \epsilon_i)$ is a monotonically decreasing function of ϵ_i,in the interval $(0, \sigma)$, from $+\infty$ to 0. Therefore, there exists some value for ϵ_i, say ϵ_i' such that $g(\epsilon_i') = 0$. Since we assume that the length of $\mathcal{C}_i$ decreases with ϵ_i, we choose the maximum value allowed for $\epsilon_i = \epsilon_i'$, which leads to the minimum-length code $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon, \sigma)$. □

We are, however, interested in the shortest-length $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon, \sigma)$ code for any value of $0 < \sigma < 1/(c+1)$. Therefore, we must perform a search in that interval.

As an example, in the figure 3.2 we have considered the following parameters: $M = 10^6$, $c = 20$ and $\epsilon = 10^{-3}$. It can be appreciated that for these parameters the optimal code occurs for $\sigma = 0.0037$, approximately.

As the equations involved in the computation of the length are transcendental, the only way to determine the optimal value for σ is through numerical simulation. We denote by $\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$ the optimal code.

3.6.2 Algebraic-geometric codes as outer codes

Finally we present another construction which is asymptotically optimal. The implementation consists of the concatenation of an outer algebraic-geometric code with an inner fingerprinting code. Recall that there exist codes approaching the TVZ bound (3.10):

$$k_o + d_o \geq n_o - n_o/(\sqrt{q} - 1).$$

These codes satisfy $n_o = O(\log(M))$. Let $\mathcal{AG}_q(n_o, k_o, d_o)$ be one of those codes satisfying (3.13). Since we are interested in positive-rate algebraic-geometric codes, i.e. $k_o/n_o > 0$, for d_o, we have that it must be an integer

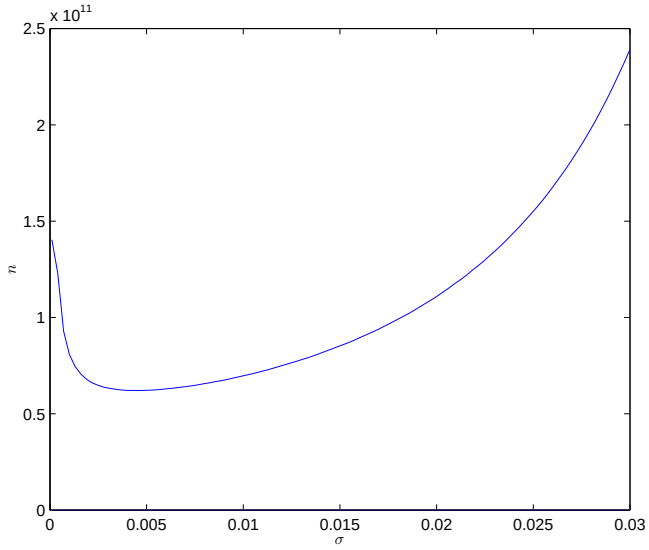


Figure 3.2: Plot of the length of $\mathcal{CF}(M, c, \epsilon, \sigma)$ versus σ

value satisfying the TVZ bound and (3.13):

$$n_o\left(1-\frac{1-\sigma(c+1)}{c^2}\right) < d_o < n_o\left(1-\frac{1}{\sqrt{q}-1}\right). \tag{3.15}$$

Thus, a sufficient condition for the existence of such value is

$$n_o\left(1-\frac{1-\sigma(c+1)}{c^2}\right) < n_o\left(1-\frac{1}{\sqrt{q}-1}\right) - 1.$$

Solving for q, we obtain

$$\sqrt{q} > 1 + \frac{c^2}{1 - c^2/n_o - (1+c)\sigma}. \tag{3.16}$$

Note that n_o is related to the design parameter ϵ, because of (3.14), as

$$n_o \geq \frac{\log_2 \epsilon_i}{\mathrm{D}(\sigma\|\epsilon_i)}.$$

Therefore, we have the following necessary lemma to guarantee the existence of the concatenated code $\mathcal{CF}_{\mathrm{AG}}(M, c, \epsilon)$.

Lemma 3.6.5. The code $\mathcal{C} = \mathcal{AG}_q(n_o, k_o, d_o) \circ \mathcal{C}_i$ exists if $\epsilon_i < \sigma < \sigma' < 1/(c+1)$, where σ' is the root of the equation

$$h(\sigma) = 1 + \frac{c^2 D(\sigma\|\epsilon_i)}{\log_2 \epsilon} - (1+c)\sigma$$

in the interval $\epsilon_i < \sigma < 1/(c+1)$.

Proof. One can easily see that $h(\sigma)$ contains a root in the interval $\epsilon_i < \sigma < 1/(c+1)$. First, note that in that interval $h'(\sigma) = (1+c)\sigma - 1$ is a negative increasing function that reaches zero for $\sigma = 1/(c+1)$, and $h''(\sigma) = c^2 D(\sigma\|\epsilon_i)/\log_2 \epsilon$ is a negative decreasing function starting at 0. Therefore, it must exist a single value σ' where both functions meet (See figure 3.3 for a numerical example). Since $h(\sigma) = h''(\sigma) - h'(\sigma)$, it changes sign at σ'. Considering (3.14) we have that $n_o \geq c^2 D(\sigma\|\epsilon_i)/\log_2 \epsilon$, and therefore, the only interval where we can ensure the existence of the code is $\epsilon < \sigma < \sigma'$, otherwise $\sqrt{q}$ would not be a positive number. Obviously, as n_o increases, $\sigma' \to 1/(c+1)$. □

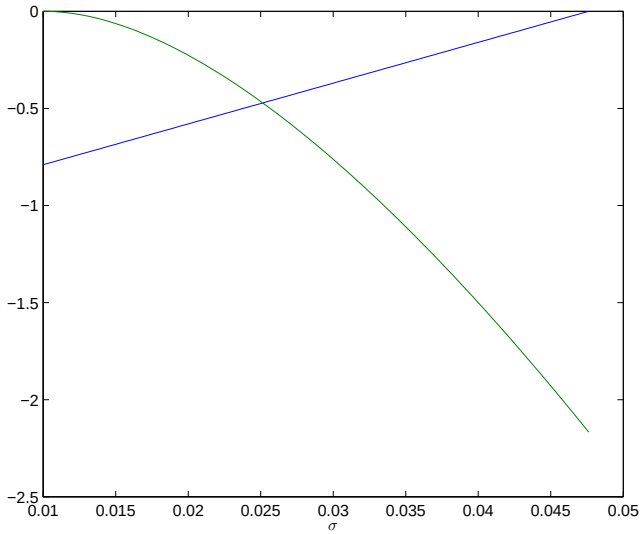


Figure 3.3: Plot of $h'(\sigma)$ and $h''(\sigma)$ for $c = 20$, $\epsilon_i = 10^{-2}$ and $\epsilon = 10^{-3}$

Asymptotically, we can suppress the term c^2/n_o from (3.16), which implies that constructible codes exist if $\epsilon_i < \sigma < 1/(c+1)$, with $q = \Omega(c^4)$. In order to obtain an efficient decoding process, in [2] it is suggested to use, as inner codes, binary c-secure Barg codes, which have been presented previously. In this case, the code has a length of order

$$O(c^6 \log \tfrac{c}{\epsilon} \log M).$$

Finally, we make some considerations about the decoding process. The inner decoding algorithm is obviously algorithm 3.3. For the outer decoding,

as in the case of the Barg codes, the Guruswami-Sudan algorithm [17] is used, which is a polynomial-complexity process. This algorithm will return a list of codewords within a radius

$$n_o - \sqrt{n_o(n_o - d_o)}.$$

If the inner code were error-free, we would only need that the outer code had $d_o > n_o - n_o/c^2$ (which would imply that it is a TA code). In this case, all the codewords in the output list would be traitors. In [2] it is shown that if $\mathbf{x}$ is a pirate codeword and $\mathbf{u}$ an innocent user of the code $\mathcal{CF}_{\mathrm{AG}}(M, c, \epsilon)$, after the inner decoding and with probability error ϵ, it is satisfied that

$$\mathrm{d}(\mathbf{u}, \mathbf{x}) \geq n_o - n_o\sigma - c(n_o - d_o) > n_o - \frac{n_o(1-\sigma)}{c}$$

Therefore, if we only want traitors in the output list, the outer code must satisfy

$$n_o - n_o\sigma - c(n_o - d_o) > n_o - \sqrt{n_o(n_o - d_o)}.$$

This implies that

$$n_o - \frac{n_o\left(\frac{1}{2} - c\sigma + \sqrt{\frac{1}{4} - c\sigma}\right)}{c^2} < d_o < n_o - \frac{n_o\left(\frac{1}{2} - c\sigma - \sqrt{\frac{1}{4} - c\sigma}\right)}{c^2} \quad (3.17)$$

This equation, together with (3.15) are the restriction of the parameters for the case of algebraic-geometric codes. After solving the system of restrictions computationally, ensuring that there exists an integer value d_o for which both conditions holds, leads to the following restrictions:

$$0 < \sigma \leq \frac{1}{(c+1)} \leq 1/2,$$

$$0 < \sigma c < \frac{1}{4}$$

and

$$q > \frac{1 - 4c\sigma + 2\sigma^2 + 2c^2\sigma^2 - 4c\sigma^3 + 2\sigma^4}{2\sigma^4} - \frac{1}{2}\sqrt{\frac{1 - 8c\sigma + 4\sigma^2 + 20c^2\sigma^2 - 24c\sigma^3 - 16c^3\sigma^3 + 4\sigma^4 + 32c^2\sigma^4 - 16c\sigma^5}{\sigma^8}}$$

Therefore, it is possible to ensure that the output list only contains traitors for sufficiently large values of q. This however has only theoretical interest in the asymptotic case. In any case, it is possible to identify a traitor, because with probability error ϵ, it is the closest codeword to the pirate fingerprint in the list returned by the Guruswami-Sudan algorithm.

3.7 Simulation results

We present in this section the simulation results. Our goal is to determine the code with the minimum length given the design parameters M, c and ϵ. As we are interested in practical implementations, we will often establish a limit around 10^7 10^8 users. The graphics show the regions of the shortest-length code.

The first simulations are focused on the different versions of the Boneh-Shaw codes: the two original constructions and the one proposed in [15]. Figure 3.4 shows the results of the simulation for these families with a fixed identification error probability $\epsilon = 10^{-10}$. For large values of M, the (random) concatenated version of the code, $\mathcal{BS}^*(M, c, \epsilon)$ has the shortest length. The reason for this is that, in the asymptotic case, the length of the non-concatenated versions are $\Omega(M)$, because they must generate M column types, one for each user.

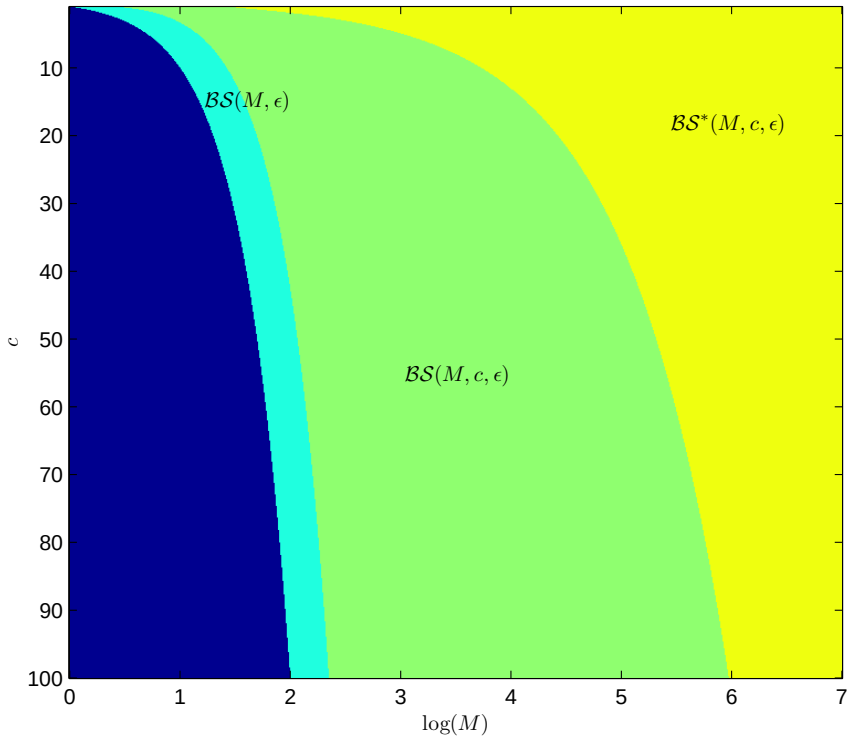


Figure 3.4: Shortest Boneh-Shaw codes

Next, we compare the Boneh-Shaw codes with binary c-SFP codes. To make this comparison, we require the same error probability for the Boneh-Shaw codes than that of the SFP codes, which is $\epsilon = 1 - 1/c$. The results are shown in figure 3.5. Since the length of SFP codes depends on a factor of the form 2^{2c}, it is disadvantageous to use them for large values of c. The simulation shows how how it is detrimental to use such codes for values $c \geq 10$.

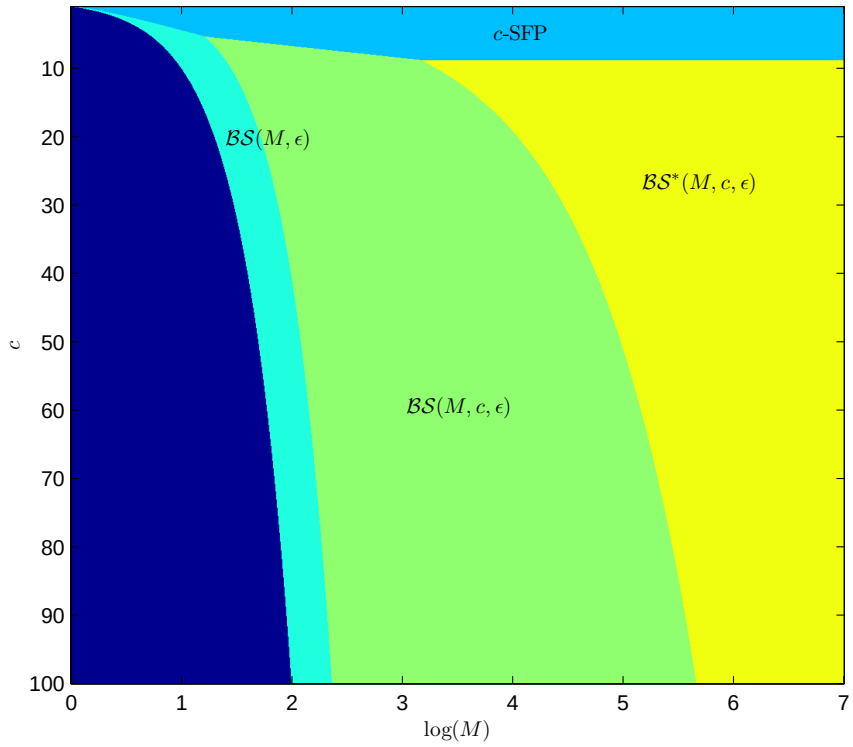


Figure 3.5: Boneh-Shaw codes versus SFP codes

In figure 3.6 we include in the simulation the Tardos codes. As we have commented previously, Tardos codes achieve the asymptotic bound for fingerprinting codes, and hence, it is not surprising that the greatest region is that corresponding to this family of codes. We will omit them in many of our simulations.

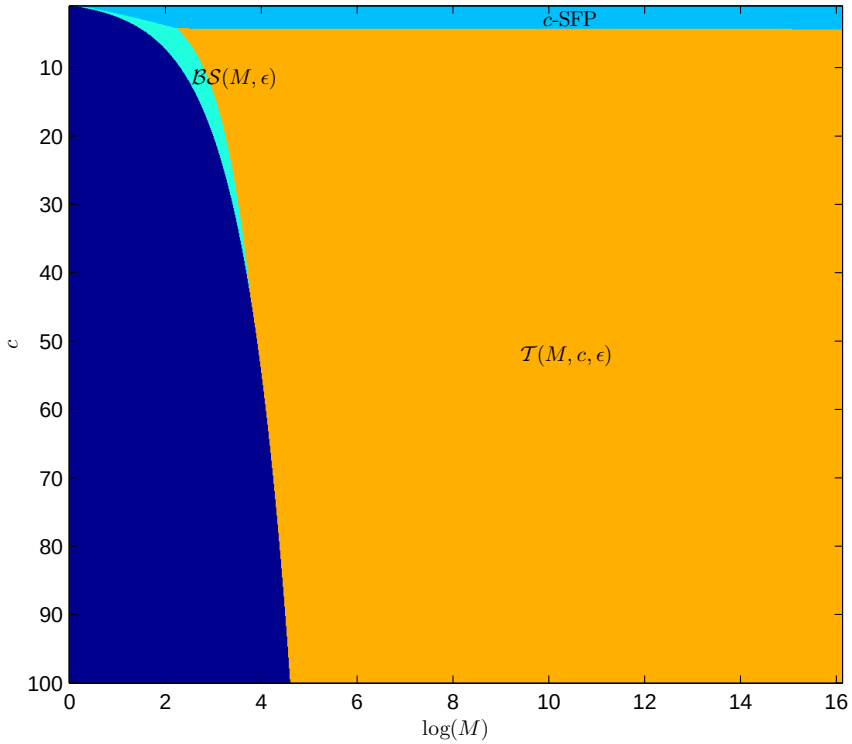


Figure 3.6: Boneh-Shaw codes versus SFP and Tardos codes

The next simulation deals with 2-secure codes. In this case, only simplex, $\mathcal{S}(k)$, and polynomial concatenated simplex codes $\mathcal{FS}(n_o, k_i)$ are, by far, the ones with the minimum length for the purposed practical scenarios, even shorter than the Tardos code. For instance, the code $\mathcal{FS}(32, 5)$ has length $n = 992$ and can allocate up to $M = 1.0995 \cdot 10^{12}$ users with $\epsilon = 1.0421 \cdot 10^{-66}$. A tardos code with the same parameters has length $n = 71863$. The sharp boundaries in the graphic are justified because the codes $\mathcal{S}(k)$ and $\mathcal{FS}(n_o, k_i)$ are parameterized, and hence, the length does not grow smoothly.

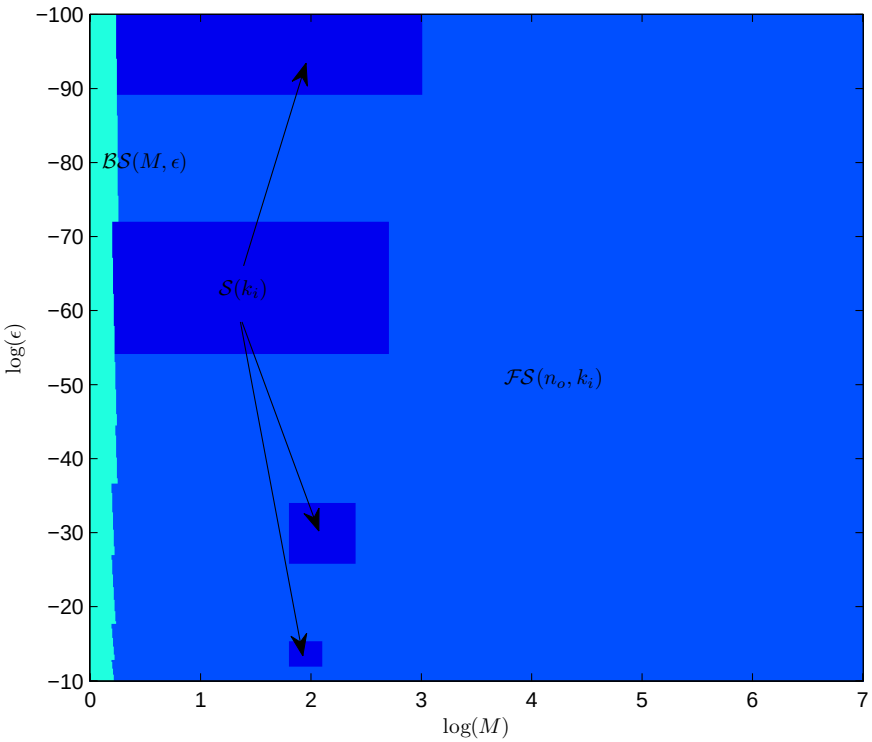


Figure 3.7: Comparison of codes for the case $c = 2$

Our next simulations deals with the quantification of the number of traitors (parameter c). Obviously, as M grows, the number of traitors also grows. However it is too pessimistic to consider that c grows linearly with M. Some authors propose $c = \log(M)$ [13]. We propose to determine c as M^a, with $0 < a \leq 1$. In the following simulations, we show how the length of the codes vary with M and ϵ.

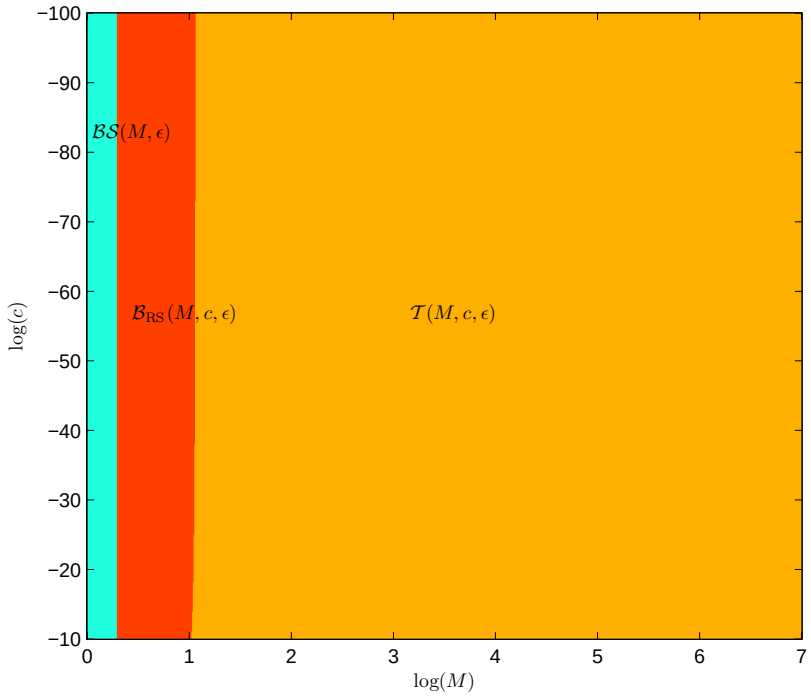
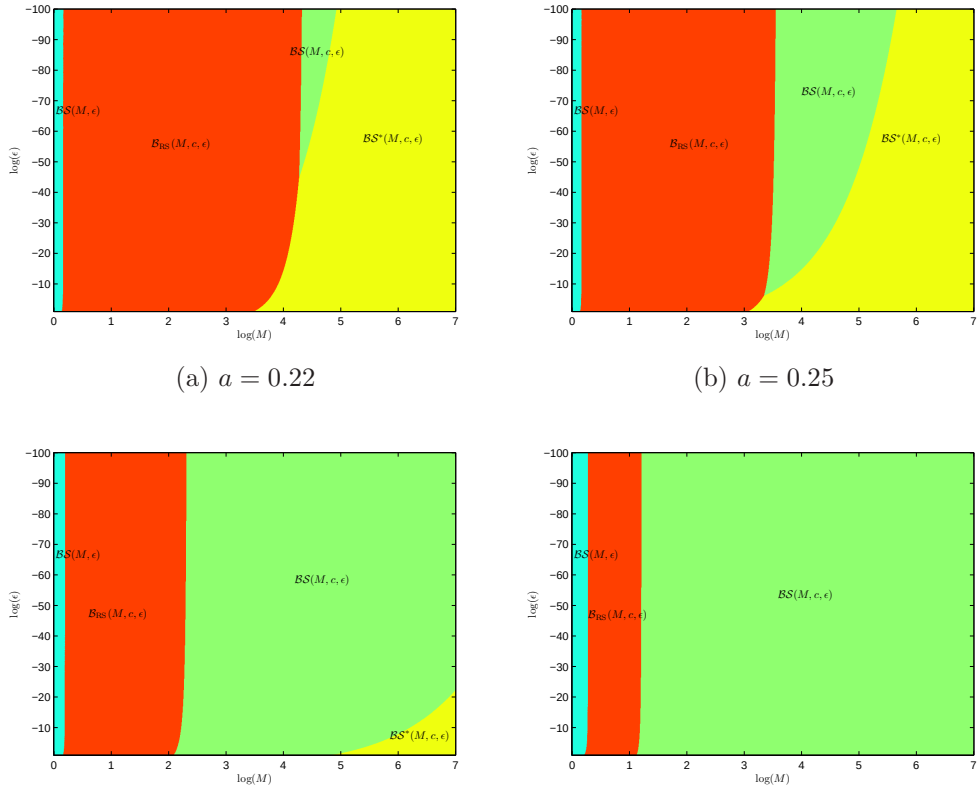


Figure 3.8: Comparison of codes for the case $c = M^a$

In figure 3.8 the Tardos almost covers the whole area. Only small strips are reserved for the Boneh-Shaw and Barg codes for small number of users. This simulation has been made with $a = 0.5$, however, little changes are appreciated irrespectively of the value of a. In the following simulations, we omit the Tardos code.

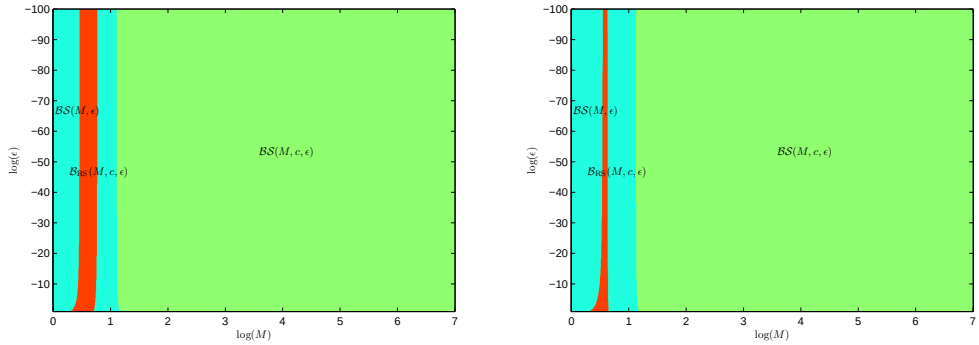
In figures 3.9 and 3.10 it can be appreciated how the areas for the minimum-length code change. Recall that the inner codes used in the Barg codes have a length that grows with 2^{2c}. That is the reason why the Barg codes are only significative for small values of α. As α grows, and hence, $c \to M$ codes designed specifically to fight against size-M becomes relevant. However, note that, even for values of a of the order of 0.95 they are only useful for $M \leq 10^6$ users, approximately.



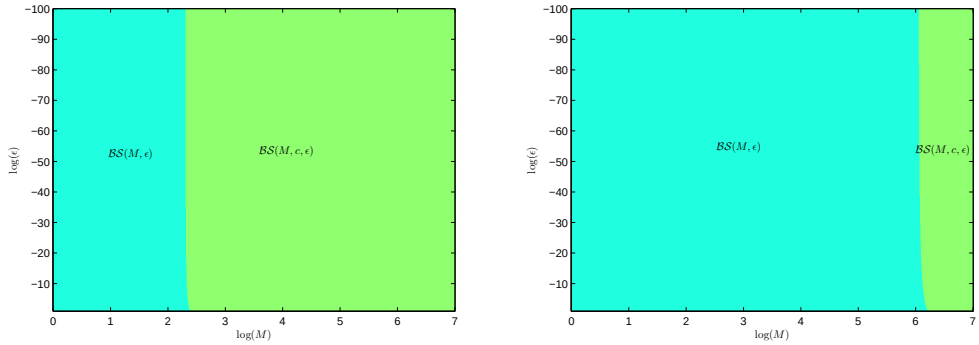
(a) $a = 0.22$ (b) $a = 0.25$

(c) $a = 0.33$ (d) $a = 0.50$

Figure 3.9: Comparison of codes for the case $c = M^a$



(a) $a = 0.59$ (b) $a = 0.60$



(c) $a = 0.85$ (d) $a = 0.95$

Figure 3.10: Comparison of codes for the case $c = M^a$

Next, we present a comparison between codes with efficient-decoding algorithms. It is a fact that randomized codes are usually better than structured codes. This has an enormous drawback, since the decoding of a random code it is known to be an NP-hard problem. The simulation has been made for the Reed-Solomon versions of the Barg and the structured concatenation of Boneh-Shaw codes. Observe how the concatenated codes follow approximately the shape of the regions in figure 3.6, which are, actually, the shape of their inner codes.

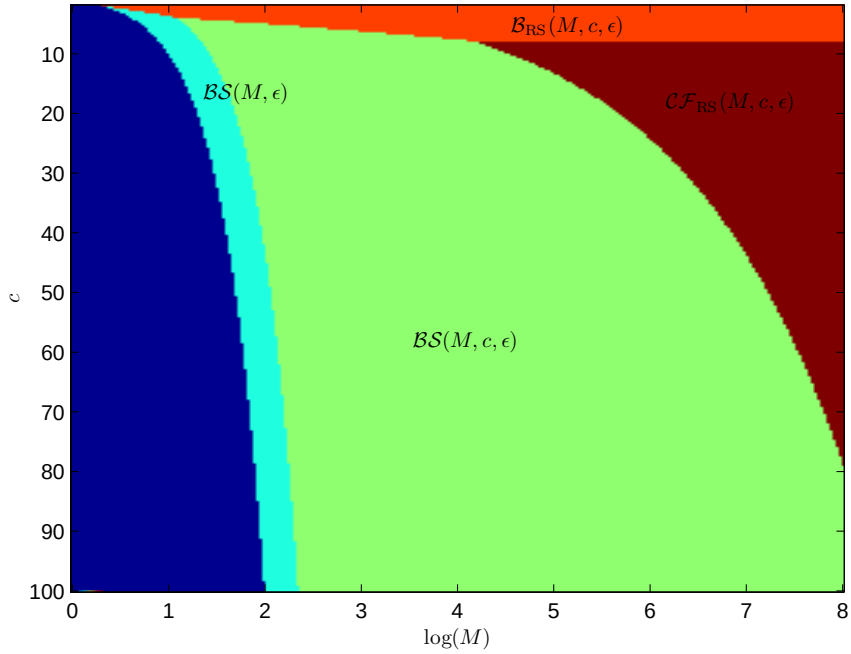


Figure 3.11: Comparison of codes with efficient-decoding algorithm.

Finally, we present some figures for some parameters of the codes, where it can be seen that randomized codes achieve usually shorter lengths.

Code	Length
$\mathcal{S}(k)$	n/a
$\mathcal{FS}(n_o, k_i)$	n/a
c-SFP	n/a
$\mathcal{BS}(M, \epsilon)$	$7.9674 \cdot 10^{22}$
$\mathcal{BS}(M, c, \epsilon)$	$3.9198 \cdot 10^{13}$
$\mathcal{BS}^{\star}(M, c, \epsilon)$	$4.9079 \cdot 10^{12}$
$\mathcal{T}(M, c, \epsilon)$	$3.9143 \cdot 10^{7}$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	∞
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	∞
$\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$	$6.4526 \cdot 10^{14}$

Table 3.1: Lengths for $M = 10^7$, $c = 100$, $\epsilon = 10^{-10}$

Code	Length
$\mathcal{S}(k)$	n/a
$\mathcal{FS}(n_o, k_i)$	n/a
c-SFP	n/a
$\mathcal{BS}(M, \epsilon)$	$8.4279 \cdot 10^{25}$
$\mathcal{BS}(M, c, \epsilon)$	$4.2905 \cdot 10^{13}$
$\mathcal{BS}^{\star}(M, c, \epsilon)$	$3.9033 \cdot 10^{10}$
$\mathcal{T}(M, c, \epsilon)$	$3.7301 \cdot 10^{7}$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	∞
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	∞
$\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$	$1.0618 \cdot 10^{12}$

Table 3.2: Lengths for $M = 10^8$, $c = 30$, $\epsilon = 10^{-10}$

Code	Length
$\mathcal{S}(k)$	$5.2700 \cdot 10^2$
$\mathcal{FS}(n_o, k_i)$	$3.7800 \cdot 10^2$
c-SFP	n/a
$\mathcal{BS}(M, \epsilon)$	$2.9190 \cdot 10^{10}$
$\mathcal{BS}(M, c, \epsilon)$	$4.7231 \cdot 10^5$
$\mathcal{BS}^\star(M, c, \epsilon)$	$1.565 \cdot 10^4$
$\mathcal{T}(M, c, \epsilon)$	$3.7301 \cdot 10^7$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	$1.759 \cdot 10^3$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	$1.6924 \cdot 10^3$
$\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$	$8.8805 \cdot 10^5$

Table 3.3: Lengths for $M = 10^7$, $c = 2$, $\epsilon = 10^{-10}$

Code	Length
$\mathcal{S}(k)$	n/a
$\mathcal{FS}(n_o, k_i)$	n/a
c-SFP	$7.7391 \cdot 10^4$
$\mathcal{BS}(M, \epsilon)$	$3.4069 \cdot 10^{22}$
$\mathcal{BS}(M, c, \epsilon)$	$6.9500 \cdot 10^{10}$
$\mathcal{BS}^\star(M, c, \epsilon)$	$2.7749 \cdot 10^6$
$\mathcal{T}(M, c, \epsilon)$	$4.0854 \cdot 10^4$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	$1.4891 \cdot 10^6$
$\mathcal{B}_{\mathrm{RS}}(M, c, \epsilon)$	$1.4414 \cdot 10^6$
$\mathcal{CF}_{\mathrm{RS}}(M, c, \epsilon)$	$3.9320 \cdot 10^7$

Table 3.4: Lengths for $M = 10^7$, $c = 5$, $\epsilon = 1 - 1/5$

4

The traceability properties of Reed-Solomon codes

As we have seen, traceable codes play a paramount role in the fingerprinting framework. Ideally, one would like to use TA codes since they can be decoded efficiently by means, for example, of a traditional half-distance decoder. Unfortunately, they are usually very long codes and require a very stringent conditions to be constructed. One may try to relax the TA condition without losing the traceability properties using IPP codes. This topic has received considerable attention in the recent years having been studied by several authors.

Given a family of codes, we are interested in *how much* can the TA property be relaxed without losing the IPP property. Specifically, in this section we are concerned with the case of Reed-Solomon codes. We try to give an answer to the question raised by Silverberg et al. in [22, 9]: Is it the case that all IPP Reed-Solomon codes are TA?. We show how often, losing the TA property implies losing more basic properties than just the IPP property.

4.1 SFP, IPP and TA codes

Recall from section 2.2 that, under the narrow-sense envelope model, a code $\mathcal{C}$ has the c-TA property if, for any pirate fingerprint generated by a c-coalition $C \subseteq \mathcal{C}$, the closest codeword of $\mathcal{C}$, in terms of the Hamming distance, is a traitor.

Recall the definition of group separation (definition 2.1.1):

$$\mathrm{D}(A, B) = |\,\mathrm{U}(A, B)|,$$

where $\mathrm{U}(A,B) = \{i : \{a_{1,i}, a_{2,i}, \dots\} \cap \{b_{1,i}, b_{2,i}, \dots\} = \emptyset\}$ is the set of separated coordinates. Recall also that we define the (c_1, c_2)-group separation of a code $\mathcal{C}$, D_{c_1,c_2}, as the minimum $D(A,B)$ for disjoint sets $A, B \subseteq \mathcal{C}$ with $|A| = c_1$ and $|B| = c_2$. For linear codes, we have the following result.

Proposition 4.1.1. For any $[n,k,d]_q$-code $\mathcal{C}$ and any pair of positive integer values c_1, c_2 it holds that

$$\max\{0, d-(c_1c_2-1)(n-d)\} \leq D_{c_1,c_2} \leq \max\{0, d-(c_1+c_2-2)(k-1)\}. \quad (4.1)$$

Proof. To prove the first inequality, let us compute the maximum number of nonseparated coordinates that any pair of disjoint sets, $A, B \subseteq \mathcal{C}$, could have. Without loss of generality, assume that $|A| = c_1$ and $|B| = c_2$, and let us call x the required number. First, note that the maximum similitude between any two codewords of $\mathcal{C}$ is $n - d$. Therefore, every codeword in A contributes, at most, with $c_2(n-d)$ nonseparated coordinates, assuming that it agrees with every codeword in B in $n-d$ coordinates. As we have c_1 codewords in A,

$$n - D_{c_1,c_2} = x \leq \min\{n, c_1c_2(n-d)\}$$

proves the inequality.

To prove the second inequality, let us construct explicitly two disjoint sets $A, B \subseteq \mathcal{C}$ in the following way. We call y the number of nondisjoint coordinates in this case. First choose two codewords $\mathbf{a}_1, \mathbf{b}_1 \in \mathcal{C}$ which agree exactly in $n-d$ coordinates. Such codewords exist by definition of the minimum distance. Insert $\mathbf{a}_1$ into A and $\mathbf{b}_1$ into B. For the d remaining coordinates were $\mathbf{a}_1$ and $\mathbf{b}_1$ do not agree, we operate in the following way. As $\mathcal{C}$ is a k-dimensional vector space, one can always find a codeword that matches any other codeword in any set of, at least, $k-1$ arbitrary coordinates.[1] Choose $c_1 - 1$ codewords such that each one of them matches $k-1$ disjoint positions with $\mathbf{b}_1$ (in the d coordinates where $\mathbf{a}_1$ and $\mathbf{b}_1$ are different) and insert them into A. Proceed similarly for the codeword $\mathbf{a}_1$ and insert the computed codewords into B. Therefore, we have that

$$n - D_{c_1,c_2} = y \geq \min\{n, (n-d) + (c_1 + c_2 - 2)(k-1)\},$$

which proves the inequality. □

Note that if we set either $c_1 = 1$ or $c_2 = 1$ in (4.1) we obtain the well-known result $n - d \geq k - 1$, which is the Singleton bound for linear codes.

[1] We omit this part of the proof.

From the previous result it follows that for a linear $[n, k, d]$-code, if $d > (c_1c_2 - 1)(n - d)$ the code is (c_1, c_2)-SFP, and if $d \leq (c_1 + c_2 - 2)(k - 1)$ the code is not (c_1, c_2)-SFP. Regarding the c-TA property we have the following sufficient condition.

Proposition 4.1.2. An (n, M, d)-code $\mathcal{C}$ is c-TA if

$$D_{1,c} > (1 - 1/c)n. \tag{4.2}$$

Proof. From the definition of $D_{1,c}$, there exists a c-coalition $C \subseteq \mathcal{C}$ that can generate a pirate codeword $\mathbf{x}$ such that $\mathrm{d}(\mathbf{x}, \mathbf{u}) = D_{1,c}$ for some $\mathbf{u} \in \mathcal{C} \setminus C$, and it does not exist any $\mathbf{u}' \in \mathcal{C} \setminus C$ such that $\mathrm{d}(\mathbf{x}, \mathbf{u}') < D_{1,c}$. Note that $\mathbf{x}$ must agree with some traitor $\mathbf{t} \in C$ at least in n/c coordinates. Therefore, $\mathrm{d}(\mathbf{x}, \mathbf{t}) \leq (1 - 1/c)n < D_{1,c} \leq \mathrm{d}(\mathbf{x}, \mathbf{u})$ for all $\mathbf{u} \in \mathcal{C} \setminus C$. □

It is not difficult to see that

$$d > (1 - 1/c^2)n \tag{4.3}$$

implies the previous condition, and therefore the code is c-TA too. The converse is, in general, not true.

Example 4.1.3. Consider the following two codes over $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$:

$$\begin{array}{ll} \mathcal{C} & = \{(0,0), \quad (1,1), \quad (\alpha,\alpha), \quad (\alpha^2,\alpha^2)\} \\ \mathcal{C}' & = \{(0,0,0), \quad (1,1,0), \quad (\alpha,\alpha,0), \quad (\alpha^2,\alpha^2,0)\} \end{array}$$

Observe that, for $c = 2$, $\mathcal{C}$ satisfies both (4.2) and (4.3), whereas $\mathcal{C}'$ only satisfies (4.2). Nevertheless, they both are 2-TA codes. For $c = 3$, $\mathcal{C}'$ does not satisfy neither (4.2) nor (4.3), even though it is a 3-TA code.

If d can be easily computed, it is often an easy way to determine or construct c-TA codes rather than computing the value $D_{1,c}$. We can now expand the diagram of equation (2.1) as follows:

$$d > (1 - 1/c^2)n \Rightarrow D_{1,c} > (1 - 1/c)n \Rightarrow c\text{-TA} \Rightarrow c\text{-IPP} \Rightarrow (c, c)\text{-SFP} \tag{4.4}$$

Codes that meet the Singleton bound, $M \leq q^{n-d+1}$, are called maximum distance separable (MDS) codes. Linear MDS codes satisfy that $n-d = k-1$ and conditions (4.2) and (4.3) are equivalent for them [23]. What is more, the following result also holds.

Theorem 4.1.4 ([23])**.** Let $\mathcal{C}$ be an $[n, k, d]_q$-code with $n \leq q + 1$. Then, for $c \geq 2$, $\mathcal{C}$ has the c-TA property if and only if $d > (1 - 1/c^2)n$.

Reed-Solomon codes are an important family of linear MDS codes which are also cyclic. We denote by $\mathbf{t}^{(i)}$ the cyclic rotation in i coordinates to the right of $\mathbf{t} \in \mathbb{F}_q^n$. In [22, 9] authors raised the question wether it is the case that all c-IPP Reed-Solomon codes are c-TA codes. Even though the conditions that appear on the right in (4.4) are more stringent than those on the left, for a large family of Reed-Solomon codes it turns out that (c, c)-SFP $\Leftrightarrow d > (1 - 1/c^2)n$. In the next section we present a method to find nonseparated configurations when the code is not c-TA and c divides the field size.

We describe here one last convention that we will use henceforth. Let f be a polynomial $f \in \mathbb{F}_q[x]$. We can define the map $f : \mathbb{F}_q \to \mathbb{F}_q$ as $x \mapsto f(x)$. We shall immediately become less formal and refer to this map simply as the polynomial f. We will be specially interested in polynomials f such that $f : \mathbb{F}_q \to \mathbb{F}_q$ is homomorphic.

4.2 Equivalence of the traceability properties of Reed-Solomon codes

The main result of this section comes in the form of the following theorem.

Theorem 4.2.1. Let $\mathcal{RS}_q(k)$ be a Reed-Solomon code over $\mathbb{F}_q$ and c a divisor of q. Then, if the minimum distance of $\mathcal{RS}_q(k)$ satisfies $d \leq n - n/c^2$ the code is not (c, c)-SFP.

The proof of the theorem can get somewhat lost in the notation and the construction of the elements that appear in it. Because of this, we first present aprocedure which summarizes the idea underneath the construction of the elements in the proof.

Let $\mathcal{RS}_q(k)$, c and d be as stated in theorem 4.2.1. Note that in this situation $\mathcal{RS}_q(k)$ is not c-TA. Our goal is to find a pair of subsets of size at most c, $C_1, C_2 \subseteq \mathcal{RS}_q(k)$, such that $\mathrm{D}(C_1, C_2) = 0$. That would prove that $\mathcal{RS}_q(k)$ fails to be (c, c)-SFP too.

Given the finite field $\mathbb{F}_q$, an integer value c satisfying $c|q$, and a Reed-Solomon code over $\mathbb{F}_q$, $\mathcal{RS}_q(k)$, with minimum distance $d \leq (1 - 1/c^2)n$, the following procedure outputs a pair of subsets $T_1, T_2 \subseteq \mathcal{RS}_q(k)$ such that they are (c, c)-nonseparated:

1. If $c^2 > q$ then:

   (a) Set $c' = \min\{c, n\}$.

(b) Find a codeword $\mathbf{t} = (t_1, \ldots, t_n) \in \mathcal{RS}_q(k)$ such that $|\mathfrak{T}| = c'$, where $\mathfrak{T} = \{t_1, \ldots, t_{c'}\}$.

(c) Return

$$T_1 = \{t\mathbf{1} : t \in \mathfrak{T}\} \text{ and}$$
$$T_2 = \{\mathbf{t}^{(ic')} : 1 \leq i \leq \lceil n/c' \rceil\}.$$

2. Else ($c^2 \leq q$):

(a) Find an additive subgroup $G \leq \mathbb{F}_q$ with q/c^2 elements.

(b) Find a nontrivial minimum-degree polynomial $f \in \mathbb{F}_q[x]$ with the elements of G as single-multiplicity roots (the application $f : \mathbb{F}_q \to \mathbb{F}_q$ will act as an additive homomorphism with $\ker f = G$).

(c) Find a subgroup $S \leq \operatorname{im} f$ of c elements and its c cosets, $\beta_1 + S, \ldots, \beta_c + S$. Set $\mathfrak{B} = \{\beta_1, \ldots, \beta_c\}$.

(d) Set $r = \text{random}\{1, \ldots, c\}$ and consider the coset $\beta_r + S$.

(e) Return

$$T_1 = \{\beta_j\mathbf{1} : \beta_j \in \beta_r + S\} \text{ and}$$
$$T_2 = \{\text{ev}(f(x) - \beta_i) : \beta_i \in \mathfrak{B}\}.$$

Note that the procedure, as well as the proof of theorem 4.2.1, is split in two cases. The first case is proved in the following proposition.

Proposition 4.2.2. Let $\mathcal{RS}_q(k)$ be a Reed-Solomon code over $\mathbb{F}_q$ and c an integer satisfying $c^2 \geq q - 1$. Then, if the minimum distance of $\mathcal{RS}_q(k)$ satisfies $d \leq n - n/c^2$ the code is not (c, c)-SFP.

Proof. According to the stated restrictions, we have that $n = q - 1$ and $k \geq 2$. This means that $\mathcal{RS}_q(2) \subseteq \mathcal{RS}_q(k)$, i.e. $\mathcal{RS}_q(k)$ contains all the codewords resulting from the evaluation of constant and linear polynomials. Take a nontrivial linear polynomial and its associated codeword, $\mathbf{t}$. Take the first $c' = \min\{c, n\}$ coordinates of $\mathbf{t}$, $\{t_1, \ldots, t_{c'}\}$, which are all different, and construct C_1 and C_2 as in step 1c of the procedure. Note that $|C_2| = \lceil n/c' \rceil \leq |C_1| = c' \leq c$. One can easily check that for the coordinates with indexes $(i-1)c' + 1, \ldots, ic'$ the codeword $\mathbf{t}^{(ic')} \in C_2$ takes values in $\mathfrak{T}$, for $1 \leq i \leq \lceil n/c' \rceil$. Because $\lceil n/c' \rceil \geq n$, for each coordinate there exist some codeword in C_2 which takes a value in $\mathfrak{T}$ and, therefore, agrees in that coordinate with some codeword in C_1. Hence, $\mathrm{D}(C_1, C_2) = 0$, which implies that $\mathcal{RS}_q(k)$ is not (c, c)-SFP. □

To prove the second case, we need the following supporting lemmas.

Lemma 4.2.3. Let R be an additive subgroup of r elements of the finite field $\mathbb{F}_q$, $R \leq \mathbb{F}_q$. Then, if m divides r there exists a subgroup $S \leq R$ with m elements.

Proof. The Sylow theorems [24] guarantee the existence of S. We show a constructive way to find S which works for the case of finite fields. Since $R \leq \mathbb{F}_q$, r must divide q. Let us convey $q = p^k$, $r = p^j$ and $m = p^i$, for some prime number p and some positive integers $k \geq j \geq i$, and let us call S_i the subgroup with p^i elements. The construction is by induction on i. For $i = 0$ simply take $S_0 = \{0\}$. For $0 < i \leq j$ take first any subgroup $S_{i-1} \leq R$ with $|S_{i-1}| = p^{i-1}$ elements and compute

$$S_i = \bigcup_{k=0}^{p-1} \left[\left(\sum_{l=1}^{k} \beta \right) + S_{i-1} \right],$$

where $\beta \in R \backslash S_{i-1}$. Using the fact that $\sum_{l=1}^{p} \beta = 0$, it is routine to check that S_i is an additive subgroup with p^i elements. Note that, in general, neither R nor S need not be isomorphic to $\mathbb{F}_r$ and $\mathbb{F}_m$, respectively. □

Lemma 4.2.4. Given the finite field $\mathbb{F}_q$ and a divisor of q, m, there exists a nontrivial polynomial $f \in \mathbb{F}_q[x]$ of degree m such that the application $f : \mathbb{F}_q \to \mathbb{F}_q$ is an additive homomorphism.

Proof. By lemma 4.2.3, take $R = \mathbb{F}_q$ and a subgroup $G \leq R$ of m elements, $G = \{g_1, \ldots, g_m\}$, and construct a polynomial having the elements of G as single-multiplicity roots,

$$f(x) = \rho \prod_{i=1}^{m} (x - g_i), \quad \rho \in \mathbb{F}_q^*.$$

Note that the polynomial $f(x)$ vanishes in all the elements of the subgroup G and the polynomial $f_\beta(x) = f(x) - f(\beta)$ vanishes in the coset $\beta + G$. This happens because f takes the same value in all the elements of $\beta + G$. Also we have that

$$\begin{aligned} f(-x) = \rho \prod_{i=1}^{m} (-x - g_i) &= \rho \prod_{i=1}^{m} (-x + g_i) \\ &= (-1)^m \rho \prod_{i=1}^{m} (x - g_i) = -\rho \prod_{i=1}^{m} (x - g_i) = -f(x). \end{aligned}$$

Note that the fourth equality is true for m odd. This is the case of any field of characteristic $\neq 2$. For fields of characteristic 2 we have that $-1 = 1$, and the equality holds too. Finally,

$$f(x+y) = \rho\prod_{i=1}^{m}(x+y-g_i) = \rho\prod_{i=1}^{m}(x-(-y+g_i))$$
$$= \rho\prod_{i=1}^{m}(x-(-y-g_i)) = f_{-y}(x) = f(x)+f(y)$$

proves that $f : \mathbb{F}_q \to \mathbb{F}_q$ it is an additive homomorphism. Note that $\ker f = G$ and $|\operatorname{im} f| = |\mathbb{F}_q/G|$. □

Now, we are in position to prove the main result of this section.

Proof of Theorem 4.2.1. We prove the theorem by explicitly finding, again, a pair of subsets which form (c,c)-nonseparated configuration.

If $c^2 > q$, the code is not (c,c)-SFP by proposition 4.2.2. From now on, we assume that $c^2 \leq q$. Under this circumstance, if c divides q, so c^2 does. Therefore, as $\mathcal{RS}_q(q/c^2+1) \subseteq \mathcal{RS}_q(k)$, it suffices to consider the case $k = q/c^2+1$. Lemma 4.2.4 provides a constructive proof of the existence of a nontrivial polynomial f of degree q/c^2 such that $f : \mathbb{F}_q \to \mathbb{F}_q$ acts as an additive homomorphism. Hence $\mathrm{ev}(f) \in \mathcal{RS}_q(k)$. Since $|\ker f| = q/c^2$, then $|\operatorname{im} f| = c^2$. Take a subgroup of c elements $S \leq \operatorname{im} f$. This can be done because $\operatorname{im} f \leq \mathbb{F}_q$ and lemma 4.2.3 guarantees the existence of S. Now, construct $\mathfrak{B}$, $\beta_r + S$, C_1 and C_2 as in steps 2c, 2d and 2e of the procedure. Note that $|C_1| = |C_2| = c$. Note also that for every $\gamma \in \mathbb{F}_q$ there exists a polynomial $f_i \in C_2$ such that

$$f_i(\gamma) \in \beta_r + S. \tag{4.5}$$

This is true because the c polynomials of C_2 replicate the c cosets of S in disjoint subsets of $\mathbb{F}_q$ due to the fact that the cosets of $\operatorname{im} f$ are disjoint. For example consider the subgroup S itself (assuming that $\beta_1 \in S$ and $r = 1$, i.e. $\beta_r + S = 0 + S$). It is replicated in the coset $\beta_2 + S$ in the polynomial $f(x) - \beta_2$, which is disjoint from S, in the coset $\beta_3 + S$ in the polynomial $f(x) - \beta_3$, which is disjoint from S and $\beta_2 + S$, etc. An analogous argument can be applied to any coset of S. Because of (4.5) and since C_1 contains the constant-valued codewords in $\beta_r + S$, an arbitrary coset, $\mathrm{D}(C_1, C_2) = 0$ and the Reed-Solomon code $\mathcal{RS}_q(k)$ is not (c,c)-SFP.

Finally, note that in the second construction we have not exploded the cyclic nature of the Reed-Solomon codes. This, together with (4.5) makes that construction be valid even for extended Reed-Solomon codes (i.e. Reed-Solomon codes where the evaluation point 0 is also considered and $n = q$). □

4.3 Example

Consider the field $\mathbb{F}_{27} = \mathbb{F}_3[x]/(x^3 + 2x + 1)$ with primitive element $\alpha = \overline{x}$. Consider the case $c = 3$ and take the Reed-Solomon code $\mathcal{RS}_{27}(4)$. First, we take the subgroup $S = \{0, 1, \alpha^{13}\}$ and construct the polynomial

$$f(x) = \rho(x-0)(x-1)(x-\alpha^{13}) = \rho(x^3 + \alpha^{13}x), \quad \rho \in \mathbb{F}_q^*.$$

For simplicity we choose $\rho = 1$. The associated codeword to f is

$$\begin{aligned}\mathrm{ev}(f) = (0, \alpha^{13}, \alpha^9, \alpha^{13}, \alpha^3, \alpha^{16}, \alpha, \alpha^3, \alpha^{22}, \alpha^{13}, \alpha, \alpha,\\ \alpha^9, 0, 1, \alpha^{22}, 1, \alpha^{16}, \alpha^3, \alpha^{14}, \alpha^{16}, \alpha^9, 1, \alpha^{14}, \alpha^{14}, \alpha^{22}),\end{aligned}$$

where it can be read off that $\operatorname{im} f = \{0,\ 1,\ \alpha,\ \alpha^3,\ \alpha^9,\ \alpha^{13},\ \alpha^{14},\ \alpha^{16},\ \alpha^{22}\}$. Since $c^2 = |\operatorname{im} f|$, we take, for example the subgroup $S = \{0, 1, \alpha^{13}\} \leq \operatorname{im} f$ of c elements and its c cosets:

$$\begin{aligned}S_1 &= \beta_1 + S = \{0, 1, \alpha^{13}\}\\ S_2 &= \beta_2 + S = \{\alpha, \alpha^3, \alpha^9\}\\ S_3 &= \beta_3 + S = \{\alpha^{14}, \alpha^{16}, \alpha^{22}\},\end{aligned}$$

where $\beta_1 = 0$, $\beta_2 = \alpha$ and $\beta_3 = \alpha^{14}$. Now consider the polynomials $f_i(x) = f(x) - \beta_i$, for $1 \leq i \leq c$. Their evaluations are those depicted in (4.6), where each coset S_i has been colored in the same way in every codeword. It can be seen that the three codewords cover disjoint positions in each coordinate for every coset. Hence, they can generate a common descendant with any of the sets whose elements are the constant words having as coordinates the elements of every coset. Valid C_1 sets are those corresponding to constant codewords in S_1, S_2 and S_3, as depicted in (4.7), (4.8) and (4.9), respectively. The common descendants are, precisely, those whose coordinates belong to the same coset S_i and are colored (underlined) in the same way.

4.4 Results for other coalition sizes

In [25] it was presented a related result proving the equivalence of SFP and TA for another families of Reed-Solomon codes. The idea there was to restate the SFP condition algebraically, as a system of equations.

Theorem 4.4.1 ([25])**.** Let $\mathcal{RS}_q(k)$ be a Reed-Solomon code over $\mathbb{F}_q$ such that $k-1$ divides $q-1$. Then, if $d \leq n - n/c^2$ the code is not (c, c)-SFP.

This covers families of codes where $c^2 \geq (q-1)/(k-1)$ with $(q-1)/(k-1)$ integer.

$$
\begin{array}{l}
(0\ ,\alpha^{13},\alpha^{9}\ ,\alpha^{13},\alpha^{3}\ ,\alpha^{16},\alpha\ \ ,\alpha^{3}\ ,\alpha^{22},\alpha^{13},\alpha\ \ ,\alpha\ \ ,\alpha^{9}\ ,0\ \ ,1\ \ ,\alpha^{22},1\ \ ,\alpha^{16},\alpha^{3}\ ,\alpha^{14},\alpha^{16},\alpha^{9}\ ,1\ \ ,\alpha^{14},\alpha^{14},\alpha^{22})\\
(\alpha^{14},\alpha^{22},1\ \ ,\alpha^{22},\alpha^{13},\alpha^{9}\ ,0\ \ ,\alpha^{13},\alpha^{3}\ ,\alpha^{22},0\ \ ,0\ \ ,1\ \ ,\alpha^{14},\alpha^{16},\alpha^{3}\ ,\alpha^{16},\alpha^{9}\ ,\alpha^{13},\alpha\ \ ,\alpha^{9}\ ,1\ \ ,\alpha^{16},\alpha\ \ ,\alpha\ \ ,\alpha^{3}\ )\\
(\alpha\ \ ,\alpha^{3}\ ,\alpha^{16},\alpha^{3}\ ,\alpha^{22},1\ \ ,\alpha^{14},\alpha^{22},\alpha^{13},\alpha^{3}\ ,\alpha^{14},\alpha^{14},\alpha^{16},\alpha\ \ ,\alpha^{9}\ ,\alpha^{13},\alpha^{9}\ ,1\ \ ,\alpha^{22},0\ \ ,1\ \ ,\alpha^{16},\alpha^{9}\ ,0\ \ ,0\ \ ,\alpha^{13})
\end{array}
\tag{4.6}
$$

$$
\begin{array}{l}
(\underline{0}\ \ ,0\ \ ,0\ \ ,0\ \ ,0\ \ ,0\ \ ,\underline{0}\ \ ,0\ \ ,0\ \ ,0\ \ ,\underline{0}\ \ ,\underline{0}\ \ ,0\ \ ,\underline{0}\ \ ,0\ \ ,0\ \ ,0\ \ ,0\ \ ,0\ \ ,\underline{0}\ \ ,0\ \ ,0\ \ ,0\ \ ,\underline{0}\ \ ,\underline{0}\ \ ,0\ \ )\\
(1\ \ ,1\ \ ,\underline{1}\ \ ,1\ \ ,1\ \ ,\underline{1}\ \ ,1\ \ ,1\ \ ,1\ \ ,1\ \ ,1\ \ ,1\ \ ,\underline{1}\ \ ,1\ \ ,\underline{1}\ \ ,1\ \ ,\underline{1}\ \ ,\underline{1}\ \ ,1\ \ ,1\ \ ,\underline{1}\ \ ,\underline{1}\ \ ,\underline{1}\ \ ,1\ \ ,1\ \ ,1\ \ )\\
(\alpha^{13},\underline{\alpha^{13}},\alpha^{13},\underline{\alpha^{13}},\underline{\alpha^{13}},\alpha^{13},\alpha^{13},\underline{\alpha^{13}},\underline{\alpha^{13}},\underline{\alpha^{13}},\alpha^{13},\alpha^{13},\alpha^{13},\alpha^{13},\alpha^{13},\underline{\alpha^{13}},\alpha^{13},\alpha^{13},\underline{\alpha^{13}},\alpha^{13},\alpha^{13},\alpha^{13},\alpha^{13},\alpha^{13},\alpha^{13},\underline{\alpha^{13}})
\end{array}
\tag{4.7}
$$

$$
\begin{array}{l}
(\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\alpha\ \ ,\alpha\ \ ,\alpha\ \ ,\underline{\alpha}\ \ ,\underline{\alpha}\ \ ,\alpha\ \ )\\
(\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\underline{\alpha^{3}}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\alpha^{3}\ ,\underline{\alpha^{3}}\ )\\
(\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\underline{\alpha^{9}}\ ,\underline{\alpha^{9}}\ ,\underline{\alpha^{9}}\ ,\alpha^{9}\ ,\alpha^{9}\ ,\alpha^{9}\ )
\end{array}
\tag{4.8}
$$

$$
\begin{array}{l}
(\underline{\alpha^{14}},\alpha^{14},\alpha^{14},\alpha^{14},\alpha^{14},\alpha^{14},\underline{\alpha^{14}},\alpha^{14},\alpha^{14},\alpha^{14},\underline{\alpha^{14}},\underline{\alpha^{14}},\alpha^{14},\underline{\alpha^{14}},\alpha^{14},\alpha^{14},\alpha^{14},\alpha^{14},\alpha^{14},\underline{\alpha^{14}},\alpha^{14},\alpha^{14},\alpha^{14},\underline{\alpha^{14}},\underline{\alpha^{14}},\alpha^{14})\\
(\alpha^{16},\alpha^{16},\underline{\alpha^{16}},\alpha^{16},\alpha^{16},\underline{\alpha^{16}},\alpha^{16},\alpha^{16},\alpha^{16},\alpha^{16},\alpha^{16},\alpha^{16},\underline{\alpha^{16}},\alpha^{16},\underline{\alpha^{16}},\alpha^{16},\underline{\alpha^{16}},\underline{\alpha^{16}},\alpha^{16},\alpha^{16},\underline{\alpha^{16}},\underline{\alpha^{16}},\underline{\alpha^{16}},\alpha^{16},\alpha^{16},\alpha^{16})\\
(\alpha^{22},\underline{\alpha^{22}},\alpha^{22},\underline{\alpha^{22}},\underline{\alpha^{22}},\alpha^{22},\alpha^{22},\underline{\alpha^{22}},\underline{\alpha^{22}},\underline{\alpha^{22}},\alpha^{22},\alpha^{22},\alpha^{22},\alpha^{22},\alpha^{22},\underline{\alpha^{22}},\alpha^{22},\alpha^{22},\underline{\alpha^{22}},\alpha^{22},\alpha^{22},\alpha^{22},\alpha^{22},\alpha^{22},\underline{\alpha^{22}})
\end{array}
\tag{4.9}
$$

Proposition 4.4.2. Given a Reed-Solomon code $\mathcal{RS}_q(k)$ over $\mathbb{F}_q$ with minimum distance d and an integer value c, if either $c|q$ or $(k-1)|(q-1)$ then, $\mathcal{RS}_q(k)$ satisfies

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

In fact, the construction presented in the step 2 in the procedure defined in section 4.2 can be easily generalized for any integer c and (extended) Reed-Solomon code over $\mathbb{F}_q$ satisfying the following property.

Proposition 4.4.3. Let $\mathcal{RS}_q(k)$ be a Reed-Solomon code over $\mathbb{F}_q$ and c an integer value satisfying

$$\sqrt{\frac{q}{\lceil q/c^2 \rceil}} \in \mathbb{N}^+. \tag{4.10}$$

Then, if the minimum distance of $\mathcal{RS}_q(k)$ satisfies $d \leq n - n/c^2$ the code is not (c,c)-SFP.

Proof. Let us call $\bar{c}$ the value of equation (4.10). It implies that $\bar{c}$ divides q. Then, by theorem 4.2.1, if $\mathcal{RS}_q(k)$ had $d \leq n - n/\bar{c}^2$ the code would not be $(\bar{c},\bar{c})$-SFP. But note that $\bar{c} \leq c$, hence $d \leq n - n/c^2 \leq n - n/\bar{c}^2$, so the code is not $(\bar{c},\bar{c})$-SFP and, obviously, not (c,c)-SFP. In other words, it exists a $\mathcal{RS}'_q(k') \subseteq \mathcal{RS}_q(k)$ such that the conditions of theorem 4.2.1 hold for a smaller value than c. The case of extended Reed-Solomon codes follows easily from this result. □

Illustratively, in table 4.1 we show some families of Reed-Solomon codes for certain values of c and q which obey (c,c)-SFP $\Leftrightarrow$ c-TA with $k = \lceil (q-1)/c^2 + 1 \rceil$. This suggests a positive answer for the question posted in [22, 9].

Table 4.1: Some families of covered Reed-Solomon codes $\mathcal{RS}_q(k = \lceil (q-1)/c^2 + 1 \rceil)$: (a) codes that satisfy $c^2 > q$, (b) codes that satisfy (4.10), (c) codes that satisfy $(k-1)|(q-1)$

$\mathbb{F}_q$	64	81	125	128	243	256	512	625	729	1024	2187
$c = 2$	(b)	(c)	(c)	(b)	–	(b)	(b)	(c)	(c)	(b)	–
3	(c)	(b)	–	–	(b)	–	–	–	(b)	–	(b)
4	(b)	(c)	–	(b)	–	(b)	(b)	(c)	–	(b)	–
5	(c)	(c)	(b)	–	–	–	–	(b)	–	–	–
8	(b)	(c)	(c)	(b)	–	(b)	(b)	–	–	(b)	–
9	(a)	(b)	(c)	(b)	(b)	(b)	(c)	(c)	(b)	–	(b)
10	(a)	(a)	(c)	(b)	(b)	(c)	–	–	(c)	(c)	–
11	(a)	(a)	(c)	(b)	(b)	(c)	–	(c)	(c)	–	–
14-15	(a)	(a)	(a)	(a)	(c)	–	–	(c)	(c)	–	–
16	(a)	(a)	(a)	(a)	(a)	(b)	(b)	(c)	–	(b)	–
17-18	(a)	(a)	(a)	(a)	(a)	(a)	(b)	(c)	–	(b)	–

$\mathbb{F}_q$	512	625	729	1024	2187
$c = 19$	(b)	(c)	–	(c)	–
20-22	(b)	(c)	(c)	(c)	–
23-24	(a)	(c)	(c)	–	–
25	(a)	(b)	(c)	–	–
26	(a)	(a)	(c)	–	–
27	(a)	(a)	(b)	–	(b)
28-31	(a)	(a)	(a)	–	(b)
32	(a)	(a)	(a)	(b)	(b)
33	(a)	(a)	(a)	(a)	(b)
34-46	(a)	(a)	(a)	(a)	(c)
≥ 47	(a)	(a)	(a)	(a)	(a)

5

Conclusion

In this project we have presented an study of the properties of the main fingerprinting codes, mainly focused on the determination of their length. We have also proposed methods to determine an estimation of this value when there is no direct method to obtain it. Specifically, we have presented and discussed some results related to SFP, Boneh-Shaw and Barg codes, and the codes presented in [2].

Also, we have presented a comparative analysis of the families of codes introduced in the project. The results show the regions where it is preferable to use each code, given the design parameters of number of users to allocate, maximum size of the collusions and identification error probability.

Finally, we have discussed the tracing properties of Reed-Solomon codes. Our main goal was to give an answer to the question posted by Silverberg et al. in [22, 9] (*Is it the case that all IPP Reed-Solomon codes are TA?*). We have proven the equivalence of the SFP, IPP and TA properties for some families of Reed-Solomon codes, where the coalition of traitors has the particularity that its size divides the field size. 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.

5.1 Future work

The main open questions are:

1. Determining a practical method to estimate the length of the $\mathcal{CF}_{\mathrm{AG}}(M, c, \epsilon)$ code.

2. Determining the limits of concatenation for fingerprinting code: when is it detrimental to concatenate in terms of code length?

3. Providing a full answer to the question about the properties of Reed-Solomon codes.

Of course, it would be interesting to determine if the last question also applies to other families of MDS or non-MDS codes.

Bibliography

[1] J. Moreira, M. Fernandez, and M. Soriano, "A note on the traceability properties of Reed-Solomon codes for certain coalition sizes," in *First IEEE workshop on information forensics and security*, 2009, (Accepted).

[2] J. Cotrina and M. Fernandez, "A family of asymptotically good binary fingerprinting codes," *IEEE Transactions on Information Theory*, 2009, (Submitted).

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

[4] M. Fernández and M. Soriano, "Fingerprinting concatenated codes with efficient identification," in *ISC '02: Proceedings of the 5th International Conference on Information Security*. London, UK: Springer-Verlag, 2002, pp. 459–470.

[5] A. Barg, G. R. Blakley, and G. A. Kabatiansky, "Digital fingerprinting codes: problem statements, constructions, identification of traitors," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 852–865, 2003.

[6] G. D. Cohen and H. G. Schaathun, "Upper bounds on separating codes," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1291–1294, 2004.

[7] 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. Comb. Theory Ser. A*, vol. 82, no. 2, pp. 121–133, 1998.

[8] A. Barg and G. Kabatiansky, "A class of i.p.p. codes with efficient identification," *Journal of Complexity*, vol. 20, no. 2-3, pp. 137–147, 2004.

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

[10] G. Cohen, S. Encheva, and H. G. Schaathun, "On separating codes," ENST, Paris, Tech. Rep., 2001.

[11] J. Moreira and M. Fernández, "Implementación y análisis de un algoritmo de decodificación de un código de fingerprinting concatenado," Universitat Politècnica de Catalunya, Tech. Rep., 2006.

[12] R. Kotter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," in *Proc. IEEE International Symposium on Information Theory*, Jun. 25–30, 2000, p. 61.

[13] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[14] B. Chor, A. Fiat, M. Naor, and B. Pinkas, "Tracing traitors," 1994.

[15] M. Fernandez and J. Cotrina, "Obtaining asymptotic fingerprint codes through a new analysis of the Boneh-Shaw codes," in *Information Security and Cryptology*, 2006, pp. 289–303.

[16] M. A. Tsfasman and S. G. Vlăduţ, *Algebraic-Geometric Codes*. Kluwer Academic Publishers, 1991.

[17] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.

[18] G. Tardos, "Optimal probabilistic fingerprint codes," *J. ACM*, vol. 55, no. 2, pp. 1–24, 2008.

[19] B. Skoric, T. U. Vladimirova, M. Celik, and J. C. Talstra, "Tardos fingerprinting is better than we thought," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3663–3676, 2008.

[20] O. Blayer and T. Tassa, "Improved versions of Tardos' fingerprinting scheme," *Des. Codes Cryptography*, vol. 48, no. 1, pp. 79–103, 2008.

[21] K. Nuida, S. Fujitsu, M. Hagiwara, T. Kitagawa, H. Watanabe, K. Ogawa, and H. Imai, "An improvement of Tardoss collusion-secure fingerprinting codes with very short lengths," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 2007.

[22] A. Silverberg, J. Staddon, and J. Walker, "Efficient traitor tracing algorithms using list decoding," in *In Proceedings of ASIACRYPT 01, volume 2248 of LNCS*, 2001, pp. 175–192.

[23] H. Jin and M. Blaum, "Combinatorial properties for traceability codes using error correcting codes," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 804–808, 2007.

[24] J. J. Rotman, *An Introduction to the Theory of Groups*. Springer-Verlag New York, Inc., 1995.

[25] M. Fernandez, J. Cotrina, M. Soriano, , and N. Domingo, "On the IPP properties of Reed-Solomon codes," in *Emerging Challenges for Security, Privacy and Trust*, ser. IFIP Advances in Information and Communication Technology, vol. 297. Springer Boston, 2009, pp. 87–97.