Wide-Sense Fingerprinting Codes and Honeycomb Arrays

Anastasia Panoui

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

2012

Wide-Sense Fingerprinting Codes and Honeycomb Arrays

Department of Mathematics
Royal Holloway, University of London

στους γονείς μου,

Μαρία και Θανάση

To my parents,

Maria and Thanasis

Declaration of Authorship

I, Anastasia Panoui, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Anastasia Panoui)

Date:

Acknowledgments

I would like to thank my supervisor Prof. Simon Blackburn for his guidance which led to the composition of this thesis. Moreover, his useful and sharp comments on my research led to the improvement and further development of my mathematical education. The pleasant environment of our research discussions and his positive view, helped me not only overcome any problems and disappointments that so often occur in research, but also to be pleased and proud of the good research results, as small or big as they might be.

I wish to thank my friends from Greece, who despite the great distance, never ceased to encourage and support me. Many thanks to my friends and fellow Ph.D. researchers who made my stay in Egham very enjoyable and the environment in McCrea building a great place to work. Special thanks to Liz Quaglia for her friendship and support and to Gaven James Watson, who was always willing to offer advice and help. I am grateful to Jean Paul Degabriele, for in stressful times he helped me view matters in a different and more relaxed perspective. I would also like to express my gratitude for the College Research Scholarship that was offered to me and made the three years of my Ph.D. studies possible.

Finally, I would like to thank my parents and my brother for their constant support and faith in me throughout the years of my studies.

Summary

The thesis is divided into two independent parts. The first part examines the main types of fingerprinting codes under four descendant models, while the second investigates the combinatorial object called a honeycomb array.

Digital fingerprinting is a technique that is used to protect intellectual rights by preventing illegal redistribution of digital data (films, music, software, etc.). This technique is facilitated by the collection of codes called fingerprinting codes. The thesis focuses on the following four fingerprinting codes: traceability, IPP, secure frameproof and frameproof. These codes are studied under four models, namely narrow-sense, expanded narrow-sense, wide-sense and expanded wide-sense. These models refer to the ability of malicious users (traitors) to produce the fingerprint in the illegal copy. In particular, following an idea of Boneh and Shaw, it is shown that there only exist trivial wide-sense traceability and IPP codes. In the matter of wide-sense frameproof codes, enhancing the relation between these codes and Sperner families first introduced by Stinson and Wei, we improve their upper bound on the size of this type of fingerprinting codes. The last two results are original and we regard the latter to be the main original contribution of this part of the thesis.

A honeycomb array of radius r is a set of $n = 2r + 1$ dots placed on the hexagonal grid in such a way that the distance of every dot from a fixed cell, the centre, is at most r. It is also required that in each column and in each diagonal only one dot occurs and that the vector differences between all pairs of dots are distinct. In the thesis it is proved that honeycomb arrays

can only be constructed using Costas arrays, which are configurations of dots in the square grid similar to honeycomb arrays. Using the existing Costas array database, all honeycomb arrays with $r \leq 14$ are determined, and two new arrays of radius 7 are presented.

Preface

This work is a composition of ideas and results from two independent areas. Thus, it is appropriate to divide the thesis into two parts. The first is called *Fingerprinting Codes*, and examines the interpretation of fingerprints into the digital world as a method of protecting intellectual property rights. The second part transfers Costas arrays to the hexagonal grid and studies the properties and the behaviour of the resulting combinatorial object, called *Honeycomb Arrays*.

Contents

List of Figures

List of Tables

Part I

Fingerprinting Codes

Chapter 1

Introduction

1.1 Protection of Intellectual Property

Digital information, whether stored in a CD, DVD or reaching the recipient through the Internet, is distributed in a continually increasing manner. However attractive is the ease of accessing and manipulating digital data, such as movies, music, software or documents, the abuse of it is tantamount to a criminal act. The protection of intellectual property is the aftermath of the invention of printing, which allowed the effortless distribution of large numbers of copies of documents and books. Today, due to the easy and direct ways of exchanging digital data, the necessity of ensuring that the rights of the creator/owner are intact is even greater. As a consequence, the research on developing methods that protect these rights is particularly important. The infringement of intellectual property rights, or in other words the illegal redistribution of data, is summarised in the word *piracy*. Digital fingerprinting, and in particular fingerprinting codes, provide a means of limiting piracy. Similar to human fingerprints which are unique for each person, the purpose of this type of code is to make identical copies of objects unique.

1.2 Outline

The aim of this section is to describe in brief the contents of each chapter of the first part of the thesis, which investigates fingerprinting codes. As the mathematical background of this type of codes is based on set theory, the

preliminary section that comprises Chapter 2 introduces the necessary definitions and notions of set theory. In particular, the statements and proofs of main results in the area of Sperner theory and the theory of intersecting sets are presented. As the name fingerprinting codes indicates, coding theory also plays a role in this topic. However, a related preliminary section on coding theory is omitted, as only the basic notions are required and are assumed to be familiar to the reader.

A detailed presentation of fingerprinting codes can be found in Chapter 3. The environment within which fingerprinting codes are used consists of the following characters: the distributor, the registered users and the traitors. The distributor is the owner of the digital data, who is responsible for its distribution to the registered users, who are the recipients of the data. The characterisation *traitors*, refers to the subset of the set of registered users who act maliciously and illegally redistribute the data. Chapter 3 describes two main applications of fingerprinting codes that are based on different settings. The first, introduced by Chor, Fiat and Naor [18], assumes that the digital data reaches the recipients through broadcast transmission, while in the second, which can be found in the survey paper by Blackburn [8], the distribution is carried out via the Internet or in a CD/DVD format. The same chapter also refers to the *marking assumption*, according to which the traitors generate fingerprints for illegal copies of the data. Moreover, four main types of fingerprinting codes are described, namely frameproof, secure frameproof, IPP and traceability codes, which correspond to different security notions, viewed from the distributor's perspective. On the other hand, taking into account the traitors' capabilities in generating the illegal fingerprint, the following four adversary models are defined: narrow-sense, expanded narrow-sense, wide-sense and expanded wide-sense descendant model. In order to provide good codes that capture both the security notion from the distributor's point of view and the adversary power, the four main types of fingerprinting codes are defined under the four aforementioned ad-

versary models, resulting in total in sixteen different types of fingerprinting codes. Lastly, Chapter 3 includes two new results. The first (Proposition 3.3.15), shows that expanded wide-sense IPP codes and totally secure codes are the same object. The second result (Propositions 3.3.22 and 3.3.25), refers to two natural ways of defining Hamming distance in the context of fingerprinting codes and proves that codes which use these two types of distance are equivalent.

The next chapter, Chapter 4, is devoted to presenting related and previous work on fingerprinting codes. For each type of code, known constructions are described and known bounds on the size of the codes are given. Chapter 5 investigates the relations amongst the sixteen different types of fingerprinting codes.

Finally, Chapter 6 is the chapter that presents the main original contribution of this part of the thesis. It focuses on the study of a particular type of fingerprinting codes, namely wide-sense frameproof codes. Using the connection between this type of codes with Sperner and intersecting families, we obtain an improvement of the known upper bound on the size of such codes (Theorem 6.3.8). Additionally, the chapter includes original results on the size of wide-sense frameproof codes of small length.

Chapter 2

Set Theory

The purpose of this chapter is to selectively present known results from set theory, that will be used in the subsequent chapters. These results relate to Sperner theory and the theory of intersecting sets.

2.1 Sperner Theory

In 1928, Sperner [46] studies the family of sets with the property that no member of the family belongs entirely to another. An important result of his study is an upper bound on the size of the maximal such family, and moreover, the type of sets that the family must contain, in order to achieve this bound.

For the remainder of the thesis, the notation n-set (accordingly n-subset), denotes a set of size n.

Definition 2.1.1 (**Sperner family**)**.** Let n be a positive integer and $\mathcal{F}$ be a family of sets over the ground set $\{1, \ldots, n\}$. The family $\mathcal{F}$ is called Sperner or has the Sperner property, if for all $A, B \in \mathcal{F}$ the sets A and B are incomparable, that is $A \not\subset B$.

Theorem 2.1.2 (Theorem 1.1.1, [26], [46])**.** *Let n be a positive integer and $\mathcal{F}$ be a Sperner family. Then*

(a)

$$|\mathcal{F}| \leq \begin{cases} \binom{n}{\frac{n}{2}}, & \text{if } n \text{ is even} \\ \\ \binom{n}{\frac{n-1}{2}}, & \text{if } n \text{ is odd.} \end{cases} \tag{2.1.1}$$

(b) Equality holds if and only if

$$\mathcal{F} = \begin{cases} \{X \subseteq \{1,\dots,n\} \,:\, |X| = \frac{n}{2}\}, & \text{if } n \text{ is even} \\ \\ \{X \subseteq \{1,\dots,n\} \,:\, |X| = \frac{n-1}{2}\} \quad \text{or} & \\ \{X \subseteq \{1,\dots,n\} \,:\, |X| = \frac{n+1}{2}\}, & \text{if } n \text{ is even.} \end{cases}$$

Proof. It is easy to see that in both the even and the odd case, the family $\mathcal{F}$ presented in the second part of the claim is Sperner and achieves the bound from part (a). Therefore, it suffices to prove that there does not exist a larger Sperner family. We first begin with the necessary notation. Let $\mathcal{F}$ be a maximal Sperner family, and define

$$l = \min\{s \,:\, \exists F \in \mathcal{F} \text{ s.t. } |F| = s\},$$
$$u = \max\{s \,:\, \exists F \in \mathcal{F} \text{ s.t. } |F| = s\}$$

to denote the size of the smallest and the largest sets in $\mathcal{F}$, respectively. Moreover, let

$$\mathcal{G} = \{X \in \mathcal{F} \,:\, |X| = l\},$$
$$\mathcal{H} = \{Y \subset \{1,\dots,n\} \,:\, |Y| = l+1 \text{ and } \exists X \in \mathcal{G} \text{ s.t. } X \subset Y\},$$
$$\mathcal{F}' = (\mathcal{F} \setminus \mathcal{G}) \cup \mathcal{H}.$$

We next prove that $\mathcal{F}'$ is Sperner. Assume for a contradiction, that it is not. Clearly, $\mathcal{F} \setminus \mathcal{G}$ is a Sperner family as a subset of $\mathcal{F}$, and $\mathcal{H}$ also satisfies the Sperner property, because it contains distinct sets of the same size. Thus, there exist $Y \in \mathcal{H}$ and $Z \in \mathcal{F} \setminus \mathcal{G}$, such that $Y \subset Z$. By the definition of $\mathcal{H}$,

there also exists $X \in \mathcal{G} \subset \mathcal{F}$, such that $X \subset Y$, which implies that $X \subset Z$, a contradiction, since both X and Z belong to $\mathcal{F}$, and $\mathcal{F}$ is Sperner family.

Let $l \leq \frac{n-1}{2}$. Also, let N be the number of pairs (X, Y) such that $X \in \mathcal{G}$, $Y \in \mathcal{H}$ and $X \subset Y$. If X is fixed, then by adding one of the elements of $\{1, \ldots, n\} \setminus X$ to X, we obtain exactly $n - l$ different sets Y that contain X. This leads to

$$N = |\mathcal{G}|(n - l).$$

On the other hand, if we fix a set Y the number of l-sets that are contained in Y is $\binom{|Y|}{l} = l + 1$, but since not all of them belong to $\mathcal{G}$, we have

$$N \leq |\mathcal{H}|(l + 1). \tag{2.1.2}$$

Combining the above results on N with $l \leq \frac{n-1}{2}$, yields

$$\frac{|\mathcal{H}|}{|\mathcal{G}|} \geq \frac{n - l}{l + 1} \geq \frac{n - \frac{n-1}{2}}{\frac{n-1}{2} + 1} = 1,$$

where equality is attained if $l = \frac{n-1}{2}$. The fact that $\mathcal{F}$ is Sperner means that $\mathcal{F} \cap \mathcal{H} = \emptyset$, thus

$$|\mathcal{F}'| = |\mathcal{F}| - |\mathcal{G}| + |\mathcal{H}| \geq |\mathcal{F}|,$$

and the equality implies $l = \frac{n-1}{2}$. Since we have chosen $\mathcal{F}$ to be the Sperner family of the maximum size, from the analysis above it is clear that the case $l \leq \frac{n-1}{2}$ leads to contradiction, because it implies the existence of a Sperner family of size greater than $\mathcal{F}$.

If $u \geq \frac{n+1}{2}$, then we also reach a contradiction. The proof of this case follows the same arguments as in the case where $\ell \geq \frac{n-1}{2}$. The only difference is that here the family $\mathcal{F}'$ is formed by replacing all u-sets S in $\mathcal{F}$ by the $(u - 1)$-sets R, such that $R \subset S$.

Hence, we can assume that $l \geq \frac{n-1}{2}$ and $u \leq \frac{n+1}{2}$. When n is even, this implies that the maximal Sperner family has size at most $\binom{n}{\frac{n}{2}}$, proving in this way the claim.

Let n be odd. If $l = u$, then

$$|\mathcal{F}| \leq \binom{n}{\frac{n-1}{2}} = \binom{n}{\frac{n+1}{2}}.$$

Hence, assume that $l = \frac{n-1}{2}$ and $u = \frac{n+1}{2}$. From the above, we have that

$$|\mathcal{F}| \leq |\mathcal{F}'| \leq \binom{n}{\frac{n+1}{2}},$$

and since $\mathcal{F}$ is maximal it must hold $|\mathcal{F}'| = |\mathcal{F}|$, which means that in (2.1.2) we have $N = |H|(l+1)$. This can occur only when all the l-subsets of a set $Y \in \mathcal{H}$ belong to $\mathcal{G}$, for all $Y \in \mathcal{H}$. Examine the sets $Y \in \mathcal{H}$ and $Z \in \mathcal{F} \setminus \mathcal{G}$, such that $|Y \cap Z|$ is the maximum. Since $\mathcal{F}$ contains only sets of size $l = \frac{n-1}{2}$ and $l+1 = \frac{n+1}{2}$, then $|Y| = |Z| = l+1$. Clearly $Y \neq Z$ and $|Y \cap Z|$ being the maximum implies that there exists $y \in Y \setminus Z$ and $z \in Z \setminus Y$. As previously mentioned $\mathcal{G}$ contains all l-subsets of any $Y \in \mathcal{H}$, which implies that $Y \setminus \{y\}$ is a member of $\mathcal{G}$. Hence, $Y' = (Y \setminus \{y\}) \cup \{z\}$ is a member of $\mathcal{H}$. However, $|Y' \cap Z| = |Y \cap Z| + 1$, which contradicts the fact that the intersection between Z and Y is maximal. Hence, in the case where n is odd, the maximal Sperner family has size $\binom{n}{\frac{n-1}{2}}$ and is attained by taking all sets of size $\frac{n-1}{2}$ or $\frac{n+1}{2}$. □

The next proposition, which can be found in the book *Sperner Theory* by Engel [26], provides upper bounds on the size of a Sperner family that depends on the sizes of the sets that it contains. The lemma that precedes the proposition presents a necessary result for the proof of the proposition.

Lemma 2.1.3 (Corollary 2.3.2, [26]). *Let $\mathcal{F}$ be a family of k-sets over a set of size $n \geq 3$ and define the families $\mathcal{H}$ and $\mathcal{D}$ as follows:*

$$\mathcal{H} = \{Y \subset \{1, \ldots, n\} \,:\, |Y| = k+1 \text{ and } \exists F \in \mathcal{F} \text{ s.t. } F \subset Y\},$$

$$\mathcal{D} = \{D \subset \{1, \ldots, n\} \,:\, |D| = k-1 \text{ and } \exists F \in \mathcal{F} \text{ s.t. } D \subset F\}.$$

(a) If $k \geq \frac{n}{2} + 1$, then $|\mathcal{D}| - |\mathcal{F}| \geq \frac{n}{2}$.

(b) If $k \leq \frac{n}{2} - 1$, then $|\mathcal{H}| - |\mathcal{F}| \geq \frac{n}{2}$.

Proposition 2.1.4 (Corollary 2.3.3, [26]). *Let $\mathcal{F} = \{F_1, F_2, \ldots, F_r\}$ be a Sperner family over the set $\{1, \ldots, n\}$ and*

$l := \min\{s \,:\, \exists i \in \{1, \ldots, r\} \text{ s.t. } |F_i| = s\}$,

$u := \max\{s \,:\, \exists i \in \{1, \ldots, r\} \text{ s.t. } |F_i| = s\}$.

If $l \leq \frac{n}{2} \leq u$, then

$$r \leq \begin{cases} \binom{n}{\lfloor \frac{n}{2} \rfloor} - (u-l)\frac{n}{2}, & \text{if } n \text{ is even,} \\ \\ \binom{n}{\lfloor \frac{n}{2} \rfloor} - (u-l-1)\frac{n}{2}, & \text{if } n \text{ is odd.} \end{cases}$$

Proof. The cases $n = 1, 2$ are trivial, so let $n \geq 3$. We prove the claim using induction on the difference $u - l$ and when n is even. The odd case can be proved analogously.

If $u - l = 0$, then we obtain the bound (2.1.1) from the previous theorem, Theorem 2.1.2. Hence, $u - l \geq 1$. For the base case $u - l = 1$ and since $l \leq \frac{n}{2} \leq u$, either $l = \frac{n}{2} - 1$ and $u = \frac{n}{2}$, or $l = \frac{n}{2}$ and $u = \frac{n}{2} + 1$. Due to the symmetry of these cases, we prove the claim when $l = \frac{n}{2} - 1$ and $u = \frac{n}{2}$. Similar to the proof of Theorem 2.1.2, we define the following:

$\mathcal{G} = \{X \in \mathcal{F} \,:\, |X| = l\}$,

$\mathcal{H} = \{Y \subset \{1, \ldots, n\} \,:\, |Y| = l + 1 \text{ and } \exists X \in \mathcal{G} \text{ s.t. } X \subset Y\}$,

$\mathcal{F}' = (\mathcal{F} \setminus \mathcal{G}) \cup \mathcal{H}$.

Since $\mathcal{F}$ consists of l-sets and $(l+1)$-sets, by replacing $\mathcal{G}$ with $\mathcal{H}$ the resulting family $\mathcal{F}'$ contains only $(l+1)$-sets. Clearly, the sets in $\mathcal{H}$ are different from the sets in $\mathcal{F} \setminus \mathcal{G}$, because otherwise the Sperner property of $\mathcal{F}$ would be violated. This implies that $\mathcal{F}'$ is also Sperner. Since $\mathcal{F} \cap \mathcal{H} = \emptyset$, applying Lemma 2.1.3 on the family $\mathcal{G}$, we obtain the following

$$|\mathcal{F}'| = |\mathcal{F}| - |\mathcal{G}| + |\mathcal{H}| \geq |\mathcal{F}| + \frac{n}{2} \quad \Rightarrow \quad |\mathcal{F}| \leq \binom{n}{\frac{n}{2}} - \frac{n}{2},$$

which proves the claim for the base case of the induction. For the inductive step we assume that

$$|\mathcal{F}| \leq \binom{n}{\frac{n}{2}} - k\frac{n}{2},$$

where $k = u - l$ and we prove that when $\mathcal{F}$ is a Sperner family for which $u - l = k + 1$, then

$$|\mathcal{F}| \leq \binom{n}{\frac{n}{2}} - (k+1)\frac{n}{2}.$$

Define $\mathcal{G}$ and $\mathcal{F}' = (\mathcal{F} \setminus \mathcal{G}) \cup \mathcal{H}$ as previously, but this time the sets in $\mathcal{F}'$ have different size. We next show that $\mathcal{F}'$ is Sperner. Assume that it is not. Clearly, $\mathcal{F} \setminus \mathcal{G}$ and $\mathcal{H}$ are both Sperner families, as a subset of $\mathcal{F}$ the former and as a family of distinct sets of the same size, the latter. Hence, there must exist a set $Y \in \mathcal{H}$ such that $Y \subset Z$, for some $Z \in \mathcal{F} \setminus \mathcal{G}$. By definition of $\mathcal{H}$, there also exists $X \in \mathcal{G}$, such that $X \subset Y \subset Z$, which is a contradiction to $\mathcal{F}$ being a Sperner family. Having removed $\mathcal{G}$ from $\mathcal{F}$, leads to the resulting family $\mathcal{F}'$ consisting of sets of size at least $l' = (l+1)$. This means that the difference between the sets of the largest and smallest size in $\mathcal{F}'$ is $u - l' = u - l - 1 = k$. By the inductive step,

$$|\mathcal{F}'| \geq \binom{n}{\frac{n}{2}} - k\frac{n}{2}. \tag{2.1.3}$$

Applying once again Lemma 2.1.3 on $\mathcal{G}$, we have that $|\mathcal{H}| - |\mathcal{G}| \geq \frac{n}{2}$. Combining this result with the bound (2.1.3) and the fact $\mathcal{F}' \cap \mathcal{H} = \emptyset$, we obtain

$$|\mathcal{F}'| = |\mathcal{F}| - |\mathcal{G}| + |\mathcal{H}| \geq |\mathcal{F}| - \frac{n}{2} \quad \Rightarrow \quad |\mathcal{F}| \leq \binom{n}{\frac{n}{2}} - (k+1)\frac{n}{2},$$

which concludes the proof. □

2.2 Intersecting Families

This section presents important results on intersecting families, an object that has captured the interest of mathematicians for many years.

Definition 2.2.1. Let $\mathcal{F}$ be a family of sets over a ground set E. Then, $\mathcal{F}$ is called *t-intersecting* if for every pair of sets $A, B \in \mathcal{F}$ we have $|A \cap B| \geq t$.

The first result we present, is the Erdős-Ko-Rado Theorem [27], that provides an upper bound on the size of intersecting Sperner families that contain sets of specific size. Apart from the result itself, the Erdős-Ko-Rado theorem plays a significant role in extremal set theory, as it introduces through its proof the method of shifting. The proof that is presented here can be found in the paper by Frankl and Graham [28], who have formalised the shifting method, and present the original proof of Erdős, Ko and Rado using this formalisation.

For convenience, when t is not specified t-intersecting families will be called intersecting. Below, we give the definition of the (i, j)-shift, followed by some properties that are needed for the proof of the Erdős-Ko-Rado theorem. The proof of these properties is omitted, as they can be easily derived from the definition of the (i, j)-shift.

Definition 2.2.2 (The $(\mathbf{i}, \mathbf{j})$-shift, [28]). Let $\mathcal{F}$ be a family of sets over $\{1, \ldots, n\}$. For $1 \leq i < j \leq n$, define

$$S_{ij}(\mathcal{F}) = \{S_{ij}(F) \,:\, F \in \mathcal{F}\},$$

where

$$S_{ij}(F) = \begin{cases} F' = (F \setminus \{j\}) \cup \{i\}, & \text{if } j \in F,\ i \notin F \text{ and } F' \notin \mathcal{F}, \\ F, & \text{otherwise.} \end{cases}$$

Proposition 2.2.3 (Proposition 2.2, [28]). *If $\mathcal{F}$ is a family of sets over $\{1, \ldots, n\}$, then*

(a) $|S_{ij}(F)| = |F|$,

(b) $|S_{ij}(\mathcal{F})| = |\mathcal{F}|$,

(c) if $\mathcal{F}$ is intersecting, then so is $S_{ij}(\mathcal{F})$.

Now we can state and prove the Erdős-Ko-Rado thoerem.

Theorem 2.2.4 (Theorem 1, [28]). *If $\mathcal{F} = \{F_1, \ldots, F_r\}$ is an intersecting Sperner family over the set $\{1, \ldots, n\}$, such that for all $i = 1, \ldots, r$ we have $|F_i| = s$ with $1 \leq s \leq \frac{n}{2}$, then*

$$r \leq \binom{n-1}{s-1}.$$

Proof. The claim is proved by induction on n. We distinguish two cases:

Case 1: $n = 2s$

Let F be a s-set over $\{1, \ldots, n\}$ and $\overline{F} = \{1, \ldots, n\} \setminus F$ be its complement. Then, all s-sets over $\{1, \ldots, n\}$ can be partitioned into $\frac{1}{2}\binom{2s}{s}$ pairs of complementary sets. Clearly, if $F \in \mathcal{F}$ then $\overline{F} \notin \mathcal{F}$. Hence,

$$|\mathcal{F}| \leq \frac{1}{2}\binom{2s}{s} = \binom{n-1}{s-1},$$

which proves the claim.

Case 2: $n > 2s$

Define $\mathcal{F}_0 = \mathcal{F}$ and for $i = 1, \ldots, n-1$ let $\mathcal{F}_i = S_{in}(\mathcal{F}_{i-1})$. According to Proposition 2.2.3(b) and (c), families $\mathcal{F}$ and $\mathcal{F}_{n-1}$ have the same size and since $\mathcal{F}$ is intersecting, $\mathcal{F}_{n-1}$ is intersecting too. Define the families $\mathcal{G}$ and $\mathcal{H}$ as follows:

$$\mathcal{G} = \{F \in \mathcal{F}_{n-1} : n \notin F\},$$
$$\mathcal{H} = \{F \setminus \{n\} : n \in F \in \mathcal{F}_{n-1}\}$$

We have that $|\mathcal{F}| = |\mathcal{G}| + |\mathcal{H}|$. By definition, $\mathcal{G}$ is an intersecting family over the set $\{1, \ldots, n-1\}$, which by induction leads to

$$|\mathcal{G}| \leq \binom{(n-1)-1}{s-1} = \binom{n-2}{s-1}.$$

We next prove that $\mathcal{H}$ is also intersecting. Assume for a contradiction, that there exist sets $H, H' \in \mathcal{H}$ such that $H \cap H' = \emptyset$. Since the size of the

sets in $\mathcal{H}$ is $s-1$, we have that $|H \cup H'| = 2(s-1) < n-1$, which implies that there exists $i \in \{1, \ldots, n-1\}$, such that $i \notin H \cup H'$. By definition, $F = H \cup \{n\}$ belongs to $\mathcal{F}_{n-1}$. Moreover, $n \in F$, which implies that through the shifting process, none of the members of F were replaced by n. In other words, for all $i \in \{1, \ldots, n-1\}$we have $S_{in}(F) = F$, which means that $(F \setminus \{n\}) \cup \{i\} = H \cup \{i\} \in \mathcal{F}_{i-1}$ and hence $H \cup \{i\} \in \mathcal{F}_{n-1}$. By assumption, $H \cap H' = \emptyset$. Thus, $(H \cup \{i\}) \cap (H' \cup \{n\}) = \emptyset$, a contradiction to the fact that $\mathcal{F}_{n-1}$ is intersecting.

As both $\mathcal{G}$ and $\mathcal{H}$ are intersecting families over $\{1, \ldots, n-1\}$, with $\mathcal{G}$ consisting of s-sets and $\mathcal{H}$ of $(s-1)$-sets, we have

$$|\mathcal{F}| = |\mathcal{G}| + |\mathcal{H}| \leq \binom{n-2}{s-1} + \binom{n-2}{s-2} = \binom{n-1}{s-1},$$

which is the desired bound.

□

The Erdős-Ko-Rado theorem refers to intersecting Sperner families that consist of sets of certain size. Results regarding the general case, where the intersecting Sperner family contains sets of arbitrary size, were provided by Milner in [37]. The proof of Milner's result is based on the following theorem by Katona, which will not be proved here.

Theorem 2.2.5 (Theorem 2, [31]). *Let $\mathcal{F}$ be a family of ℓ-sets over $\{1, \ldots, n\}$, that is k-intersecting. Let $1 \leq g \leq \ell$, $1 \leq k \leq \ell$, $g + k \geq \ell$ and*

$\mathcal{B} = \{B : |B| = g \text{ and } B \subset F, \text{ for some } F \in \mathcal{F}\}$.

Then,

$$|\mathcal{B}| \geq \frac{\binom{2\ell-k}{g}}{\binom{2\ell-k}{\ell}} |\mathcal{F}|,$$

where strict inequality holds in the following two cases:

(a) $g = \ell$

(b) $\mathcal{F}$ *consists of all* ℓ*-sets over the set* $E \subset \{1, \ldots, n\}$ *of size* $|E| = 2\ell - k$.

Theorem 2.2.6 (Theorem 1, [37]). *If* $\mathcal{F} = \{F_1, \ldots, F_r\}$ *is a* k*-intersecting Sperner family over the set* $\{1, \ldots, n\}$*, then*

$$r \leq \binom{n}{\lfloor \frac{n+k+1}{2} \rfloor}.$$

Proof. Let $t = \lfloor \frac{n+k+1}{2} \rfloor$. Clearly, if all sets in $\mathcal{F}$ have size exactly t, then the claim is trivially true. We consider two cases, depending on the sizes of the sets that comprise the family $\mathcal{F}$.

Case 1: For all $i \in \{1, \ldots, r\}$, $|F_i| \leq t$ and there exists some i, such that $|F_i| < t$.

In other words, if ℓ is the size of the smallest set in $\mathcal{F}$, then we assume that for $1 \leq s \leq n$

$$\ell = |F_1| = \ldots = |F_s| < |F_{s+1}| \leq \ldots \leq |F_r| \leq t.$$

For the sets $F_1, \ldots, F_s$, let $C_1, \ldots, C_q$ be the distinct $(\ell + 1)$-sets, such that for some $i \in \{1, \ldots, s\}$ and $j \in \{1, \ldots, q\}$ we have $F_i \subset C_j$. Then, since for every $i, j \in \{1, \ldots, r\}$ $|F_i \cap F_j| \geq k$, it also holds that $|C_i \cap F_j| \geq k$, for all $i \in \{1, \ldots, q\}$ and $j \in \{s + 1, \ldots, r\}$, which means that the family $\{C_1, \ldots, C_q, F_{s+1}, \ldots, F_r\}$ is k-intersecting. Furthermore, if there exist $i \in \{1, \ldots, q\}$ and $j \in \{s + 1, \ldots, r\}$ such that $C_i \subseteq F_j$, then $F_i \subset F_j$, which violates the Sperner property of $\mathcal{F}$. Hence, the sets $C_1, \ldots, C_q, F_{s+1}, \ldots, F_r$ form a k-intersecting Sperner family of size $q + r - s$.

For a set A, let $\overline{A} = \{1, \ldots, n\} \setminus A$ denote the complement of A. Then, $|\overline{C}_i| = n - (\ell + 1)$ and there exists $i \in \{1, \ldots, q\}$, such that $\overline{C}_i \subset \overline{F}_j$, for $j \in \{1, \ldots, s\}$. Also, for every $i, j \in \{1, \ldots, s\}$

$$|\overline{F}_i \cap \overline{F}_j| = |\overline{F_i \cup F_j}| = n - (|F_i| + |F_j| - |F_i \cap F_j|) \geq n + k - 2\ell.$$

Notice, that by assumption $\ell \leq t-1$ and hence $n+k-2\ell \geq 1$. Applying Katona's theorem (Theorem 2.2.5) on $\{\overline{F}_1, \ldots, \overline{F}_s\}$, we have

$$q \;\geq\; \frac{\binom{2(n-\ell)-(n+k-2\ell)}{n-\ell-1}}{\binom{2(n-\ell)-(n+k-2\ell)}{n-\ell}}\, s \;=\; \frac{\binom{n-k}{n-\ell-1}}{\binom{n-k}{n-\ell}} s \;\geq\; s, \qquad (2.2.1)$$

where the last inequality holds because $n + k - 2\ell \geq 1$.

If $\ell = t - 1$, then for all $\{1, \ldots, q\}$ we have $|C_i| = t$, which leads to $\{C_1, \ldots, C_q, F_{s+1}, \ldots, F_r\}$ being a k-intersecting Sperner family of t-sets and size $q + r - s$. Using (2.2.1) we obtain the desired bound for r.

If $\ell < t - 1$, then the claim is proved using induction on $t - \ell$.

Case 2: There exists some $i \in \{1, \ldots, r\}$, such that $|F_i| > t$.

In this case, the family $\mathcal{F}$ consists of sets with different sizes, hence we can assume the following:

$$|F_1| \leq \ldots \leq |F_s| < t \leq |F_{s+1}| \leq \ldots \leq |F_p| < |F_{p+1}| = \ldots = |F_r| = \ell',$$

where $1 \leq s \leq p < r$. Let $D_1, \ldots, D_q$ be the distinct t-sets, for which there exists $i \in \{1, \ldots, s\}$ such that $F_i \subset D_j$, for some $j \in \{1, \ldots, q\}$. Then, following the same argument as in the previous case we obtain

$$q \geq s. \qquad (2.2.2)$$

Let $B_1, \ldots, B_u$ be the distinct $(l' - 1)$-sets, such that there exists an $i \in \{p + 1, \ldots, r\}$ for which $B_j \subset F_i$, for all $j \in \{1, \ldots, u\}$. Next, for $j \in \{1, \ldots, u\}$ and $i \in \{p+1, \ldots, r\}$ we count in two ways the number N of pairs (B_j, F_i) such that $B_j \subset F_i$. Recall, that $|B_j| = l' - 1$ and $|F_i| = l'$. For fixed j, there exist exactly $n - (l' - 1)$ sets F_i, such that $B_j \subset F_i$, and since we have u sets B_j we get

$$N = u(n - l' + 1). \qquad (2.2.3)$$

On the other hand, if we fix an i, the number of $(l'-1)$-subsets of F_i is l', but as not all of them belong to $\{B_1, \ldots, B_u\}$, we have

$$N \leq (r-p)l',$$

which combined with (2.2.3) yields

$$u(n-l'+1) \geq (r-p)l'. \tag{2.2.4}$$

As $l' > t$, we have that

$$2\ell' > n+k+1 > n+1, \tag{2.2.5}$$

since $\mathcal{F}$ is intersecting and thus $k > 0$. Inequality (2.2.5) implies that

$$\frac{\ell'}{n-\ell'+1} > 0$$

and so (2.2.4) gives

$$u > r-p. \tag{2.2.6}$$

It is easy to check that the sets $D_1, \ldots, D_q, F_{s+1}, \ldots, F_p, B_1, \ldots, B_u$ form a k-intersecting Sperner family. If $l' = t+1$, then this family consists of sets of size $l'-1$ and l', and has size $q+(p-s)+u$. By Case 1, the size of the family is

$$q+(p-s)+u \leq \binom{n}{\ell'-1} = \binom{n}{t}.$$

Combining inequalities (2.2.1) and (2.2.6) we have

$$r = q+(p-s)+u \leq \binom{n}{t}$$

and the claim is proved.

If $l' > t+1$, then the desired bound is obtained by applying an induction argument on $l'-t$.

□

A different approach on the study of intersecting Sperner families, was introduced by a conjecture made by Purdy and proven true by Schonheim [44]. Instead of looking at intersecting Sperner families, the conjecture considers Sperner families with the property that the union of any two distinct pairs of sets, does not cover the ground set.

Definition 2.2.7. Let $\mathcal{F}$ be a family over a ground set E. Then $\mathcal{F}$ is called *non 2-covering* if for every pair of sets $A, B \in \mathcal{F}$ we have $A \cup B \neq E$.

Theorem 2.2.8 ([44]). *Let $\{1, \ldots, n\}$ be a set of even size. If $\mathcal{F} = \{F_1, F_2, \ldots, F_r\}$ is a non 2-covering Sperner family over $\{1, \ldots, n\}$, then*

$$r \leq \binom{n}{\frac{n}{2} - 1}.$$

Proof. Let $B_1, \ldots, B_s$ be sets in $\mathcal{F}$ that have size greater than $\frac{n}{2}$. According to a result in [20] by De Bruijn, Tengbergen and Kruijswijk, the set of all subsets of $\{1, \ldots, n\}$ can be decomposed into pairwise disjoint symmetric chains. Since $\mathcal{F}$ is Sperner, $B_1, \ldots, B_s$ belong to different chains. Based on the result of De Bruijn et al., for $i = 1, \ldots, s$ the set B_i can be replaced by the $\frac{n}{2}$-set C_i from the chain containing B_i. Then, since $C_i \subset B_i$, the condition $B_i \cup B_j \neq \{1, \ldots, n\}$ implies that $C_i \cup C_j \neq \{1, \ldots, n\}$. Furthermore, if F is a set in $\mathcal{F}$ of size less than $\frac{n}{2}$, then it also holds that $C_i \cup F \neq \{1, \ldots, n\}$, since $|C_i \cup F| < 2\frac{n}{2} = n$. Hence, this replacement does not destroy the property of the initial family, that the union of any two sets do not cover the ground set $\{1, \ldots, n\}$.

Let $\overline{C}_i = \{1, \ldots, n\} \setminus C_i$ be the complement of C_i, for all $i = 1, \ldots, s$. Then $\overline{C}_i$ is also a $\frac{n}{2}$-set, and since $\mathcal{F}$ cannot contain complementary sets, we have

$$s \leq \frac{n}{2}\binom{n}{\frac{n}{2}},$$

which is smaller than or equal to $\binom{n}{\frac{n}{2}-1}$. This proves the claim in the case where $\mathcal{F} = \{B_1, \ldots, B_s\}$.

Consider again the decomposition into pairwise disjoint chains and let $D_1, \ldots, D_t$ be sets of size less than $\frac{n}{2}$. Then, since $\mathcal{F}$ is Sperner, $D_1, \ldots, D_t$ belong to different chains. There are $\binom{n}{\frac{n}{2}-1}$ disjoint chains that contain sets of size less than $\frac{n}{2}$, hence

$$t \leq \binom{n}{\frac{n}{2}-1},$$

and the claim is proved when $\mathcal{F} = \{D_1, \ldots, D_t\}$.

We now consider the general case. If X is a $(\frac{n}{2}-1)$-subset of a $\frac{n}{2}$-set which belongs to $\mathcal{F}$, then since $\mathcal{F}$ is Sperner, no set from the chain containing X, belongs to $\mathcal{F}$. Clearly, the $\frac{n}{2}$-sets $C_1, \ldots, C_s$ are pairwise intersecting, since are members of $\mathcal{F}$. Applying Theorem 2.2.5, we get that the number of sets X is at least

$$\frac{\binom{2\frac{n}{2}-1}{\frac{n}{2}-1}}{\binom{2\frac{n}{2}-1}{\frac{n}{2}}} s = \frac{\binom{n-1}{\frac{n}{2}-1}}{\binom{n-1}{\frac{n}{2}}} s = s.$$

In total, we have that

$$|\mathcal{F}| = s + t \leq s + \binom{n}{\frac{n}{2}-1} - s = \binom{n}{\frac{n}{2}-1},$$

which completes the proof. □

Chapter 3

The Fingerprinting Problem

This chapter introduces the notion of fingerprinting and the motivation behind the research of fingerprinting codes. In particular, two applications of fingerprinting are presented, followed by the definition of the four adversay models that describe the capabilities of the traitors: narrow-sense, expanded narrow-sense, wide-sense and expanded wide-sense model. Next, four main types of fingerprinting codes are described, namely frameproof, secure frameproof, IPP and traceability codes, which correspond to four different security notions. The combination of these security notions with the adversary models gives rise to the definition of sixteen types of fingerprinting codes. Furthermore, an original result regarding expanded wide-sense IPP codes (Proposition 3.3.15) proves that this type of codes is equivalent to the type of fingerprinting codes called totally secure codes. The chapter concludes with a new result on traceability codes, involving two natural ways of defining the Hamming distance. Propositions 3.3.22 and 3.3.25 show that traceability codes under the two different types of distance are in fact equivalent.

3.1 Digital Fingerprinting and Applications

The ease of access to the vast collection of digital data that is provided by the Internet, as well as by other means of exchanging data, requires the use of methods that prevent the illegal distribution of the data, known as *piracy*. Just as human fingerprints make each one of us unique and constitute a way of identification, digital fingerprints give the property of distinctness

amongst the copies of the digital data. The idea of fingerprinting is not new. As mentioned in [15] by Boneh and Shaw, hundreds of years ago mathematicians used this technique in the logarithm tables. In order to make each copy unique, they altered the least significant digit of randomly chosen values of $\log x$, so that each copy had a different set of $\log x$ values altered. In this way, once an illegal copy of the tables was found, it was possible to identify the traitorous owner.

Digital fingerprinting is a method of personalising digital data, such as music, films, documents, software, in order to eliminate illegal redistribution (*piracy*), by tracing the malicious users (*traitors*). Watermarking technologies provide another way of marking digital data and in some cases are regarded to be the same as digital fingerprinting. However, in the present content we consider watermarking to be a method of indicating the owner/creator of the object, whereas fingerprinting serves the purpose of detecting the malicious users.

For a better understanding of the notion of digital fingerprinting, two main applications are presented here, taken from the survey paper [8] by Blackburn. The first was introduced in 1994, by Chor, Fiat and Naor in [18]. According to their model, the data to be distributed reaches the registered users through broadcast transmission. This implies that the data can also be received by unregistered users, since there is no way of controlling broadcast signals. To avoid this situation from occurring, the distributor applies cryptographic techniques and instead of transmitting the clear data, he transmits its encrypted form. For the encryption, it is necessary that the distributor creates a base set of keys S and a set Q of q marks. Then, he divides the data into blocks and each block m, into ℓ segments. Next, he randomly chooses two sets of keys from S, namely a set $\{s_1, \ldots, s_\ell\}$ of ℓ keys and a set of $q\ell$ keys, as presented in Table 3.1. The first set of keys, is used by the distributor in constructing the session key $s = s_1 \oplus \ldots \oplus s_\ell$, while the second set, in encrypting the keys $s_1, \ldots, s_\ell$ (Table 3.2) in the following way: every

$$\begin{array}{cccc} k_{1,1} & k_{1,2} & \cdots & k_{1,\ell} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,\ell} \\ \vdots & \vdots & \ddots & \vdots \\ k_{q,1} & k_{q,2} & \cdots & k_{q,\ell} \end{array}$$

Table 3.1: The set of $q\ell$ keys that are used to encrypt keys $\{s_1, \ldots, s_\ell\}$.

$$\begin{array}{cccc} E_{k_{1,1}}(s_1) & E_{k_{1,2}}(s_2) & \cdots & E_{k_{1,\ell}}(s_\ell) \\ E_{k_{2,1}}(s_1) & E_{k_{2,2}}(s_2) & \cdots & E_{k_{2,\ell}}(s_\ell) \\ \vdots & \vdots & \ddots & \vdots \\ E_{k_{q,1}}(s_1) & E_{k_{q,2}}(s_2) & \cdots & E_{k_{q,\ell}}(s_\ell) \end{array}$$

Table 3.2: The encryptions in the enabling block, where ℓ denotes the number of segments and q the number of marks.

key s_j is encrypted under keys $k_{i,j}$, for all $i = 1, \ldots, q$. Finally, the distributor first transmits the encryptions of the keys $s_1, \ldots, s_\ell$ (*enabling block*), and next the encryption of the data block m under the session key s (*cipher block*).

At the receivers' end, it is clear that only the users who possess the keys used for the encryption, are able to decrypt the transmitted data. Hence, in order for the registered users to obtain the clear data, the distributor provides each one of them with a smartcard, which the users use as input to a decoder box. Each smartcard contains a different set of ℓ keys, that allow the users to decrypt the encrypted message. Specifically, the set consists of one key from each column of the table of keys $k_{i,j}$ (Table 3.1). In this way, all authorised users are able to decrypt the received $E_{i,j}(s_j)$, since the encryption was carried out using keys $k_{i,j}$, for all $i = 1, \ldots, q$ and $j = 1, \ldots, \ell$. Finally, the decryption of the cipher block $E_s(m)$ is now possible, by constructing the key $s = s_1 \oplus \ldots \oplus s_\ell$.

With regard to the actions of the traitors, we consider two cases. The

first case involves only one traitor, whereas the second a coalition, which is formed by at most t authorised users. The case of a sole traitor is easily addressed, as his smartcard uniquely identifies him. On the other hand, the case of a coalition is more complex. The traitors are aiming to help an unauthorised user (pirate) create an illegal smartcard that will decrypt the transmitted encrypted data. In order to achieve successful decryption, the traitors must give away the keys from their smartcards. Since the smartcards uniquely identify the users, to avoid being captured, the traitors load the pirate smartcard with a combination of their keys. This shuffle of keys from different users result in breaking the connection between them and the new created smartcard.

Another application of fingerprinting, which can also be found in the survey paper by Blackburn [8], is the case where the digital data is distributed through the Internet or in a CD/DVD form. In this scenario, the users do not receive the same copies of the data, but copies which are marked differently. Let us consider the case where the creator/owner of the data, distributes copies of a film written on DVD, to the registered users. The distributor, associates each user with a codeword chosen from a code C of length ℓ. Next, he marks the copy with this codeword and sends the marked DVD to the user. We call the collection of marks in each copy, a fingerprint. In this way, each user corresponds to a different copy of the film, which uniquely identifies him. This implies that if only one user acts traitorously and redistributes his copy, then the illegal DVD trivially indicates him as a traitor.

In the case where a coalition is formed, the traitors can hide their identity by combining the different fingerprints that are embedded in their copies. By the construction, the marks are imperceptible and hence the only way to detect their presence is to examine the differences amongst the copies of the coalition. This means that in the pirate copy, the traitors have the freedom to modify the positions where their copies differ. In particular, they can apply

one of the following modifications in these detectable positions:

(a) use the values of the corresponding positions from their copies,

(b) use the values of the corresponding positions from their copies, delete the value or turn it unreadable,

(c) use arbitrary values that comply with the alphabet that is used,

(d) use arbitrary values that comply with the alphabet that is used, delete the value or turn it unreadable.

For the previous application it is clear that the only option for constructing the illegal fingerprint is (a), as the fingerprint plays the role of the cryptographic key. Moreover, even if the fingerprint does not have key properties, method (a) could be applied by the traitors in the case where it is impossible to detect the positions of the marks. Thus, the coalition can only combine their fingerprints, without being able to modify them. In the case where the marks are partially detected, the traitors could apply the model (b) and remove the fingerprints from the known positions. In opposition to the first application, in the example with the DVD distribution, it is not essential for the traitors to use the values from their fingerprinted copies. If the marks are visible, but for some reason their extraction is either not feasible or it would cause quality degrade of the data, then in order to hide their identity the members of the coalition would modify the fingerprints by changing their value (model (c)). Finally, when the fingerprints are visible and (partially) removable, model (d) provides the best strategy, as it creates a pirate copy whose fingerprint is as unconnected to the coalition's fingerprints as possible.

3.2 The Descendant Set

Both applications mentioned above, indicate that the traitors' capability is restricted. This restriction is described by the Marking Assumption:

the members of the coalition can only alter those coordinates of the fingerprint in which at least two of their fingerprints differ,

as stated in [4]. To summarise, the fingerprinting problem focuses on the construction of a code C with the property that the distributor is always able to identify at least one member of the coalition, which has size at most t.

Let Q denote for the remainder of the first part of the thesis, an alphabet of size q. Also, let $U = \{u^1, \ldots, u^t\}$ denote the set of traitors and $D = \{\mathbf{y}^1, \ldots, \mathbf{y}^t\}$ the fingerprints that correspond to each u^i, for all $i = 1, \ldots, t$. The set of the illegal fingerprints is called the descendant set, as all these fingerprints derive from the fingerprints of the coalition. However, in the literature the descendant set could be found under the name of envelope [4] or feasible set [50, 15].

There are four different adversary models to create the descendant set that correspond to the four options (a)-(d) above: the narrow-sense, the wide-sense and their expanded versions. The first, the narrow-sense descendant set, denoted by $\mathrm{desc}(D)$, is defined as the set of all $\mathbf{x} \in Q^\ell$ which are generated using letters only from the codewords in D:

$$\mathrm{desc}(D) = \{\mathbf{x} \in Q^\ell \,:\, x_i \in \{y_i^1, \ldots, y_i^t\}\}.$$

The wide-sense descendant set, denoted $\mathrm{wdesc}(D)$, allows the traitors to substitute the marks that they are able to detect by any mark of the alphabet Q: $\mathbf{x} \in \mathrm{wdesc}(D)$ if and only if $\mathbf{x} = x_1 \ldots x_\ell$, where

$$\begin{cases} x_i = y_i^1, & \text{if } y_i^1 = y_i^2 = \ldots = y_i^t \\ x_i \in Q, & \text{otherwise.} \end{cases}$$

Each one of the defined descendant sets can be extended by introducing the symbol '?', which represents deletion or an unreadable mark. The expanded narrow-sense and expanded wide-sense descendant sets, denoted $\mathrm{desc}^*(D)$ and $\mathrm{wdesc}^*(D)$ accordingly, are defined as follows:

$\mathrm{desc}^*(D)$**:** $\mathbf{x} \in \mathrm{desc}^*(D)$ if and only if $\mathbf{x} = x_1 \dots x_\ell$, where

$$\begin{cases} x_i = y_i^1, & \text{if } y_i^1 = y_i^2 = \ldots = y_i^t \\ x_i \in \{y_i^1, \ldots, y_i^t\} \cup \{?\}, & \text{otherwise.} \end{cases}$$

$\mathrm{wdesc}^*(D)$**:** $\mathbf{x} \in \mathrm{wdesc}^*(D)$ if and only if $\mathbf{x} = x_1 \dots x_\ell$, where

$$\begin{cases} x_i = y_i^1, & \text{if } y_i^1 = y_i^2 = \ldots = y_i^t \\ x_i \in Q \cup \{?\}, & \text{otherwise.} \end{cases}$$

The example that follows describes in a concrete way the definition of the descendant sets.

Example 3.2.1. Let $D = \{10021, 20221, 20021\}$ be a subset of a code C over $Q = \{0, 1, 2\}$. Then

$$\mathrm{desc}(D) = \{10021, 10221, 20021, 20221\},$$

$$\mathrm{desc}^*(D) = \{10021, 10221, 10?21, 20021, 20221, 20?21, ?0021, ?0221, ?0?21\},$$

$$\mathrm{wdesc}(D) = \{00021, 00121, 00221, 10021, 10121, 10221, 20021, 20121, 20221\},$$

$$\mathrm{wdesc}^*(D) = \{00021, 00121, 00221, 00?21, 10021, 10121, 10221, 10?21, 20021,$$
$$20121, 20221, 20?21, ?0021, ?0121, ?0221, ?0?21\}.$$

3.3 Fingerprinting Codes

This section presents the four main types of fingerprinting codes, namely frameproof, secure frameproof, identifying parent property (IPP) and traceability codes.

3.3.1 Frameproof Codes

As mentioned in both applications, if there is only one authorised user who acts traitorously and redistributes his copy, then the illegal copy will identify him, since it bears his fingerprint. However, it is possible that the traitor pretends to be innocent and asserts that he has been framed. Therefore,

it is necessary for the distributor to be able to identify the true traitor and prevent the capture of innocent users. In 1995, Boneh and Shaw [15] introduced a new notion of fingerprinting, which possesses exactly this property. Instead of concentrating on tracing at least one traitor, the aim is to prevent the members of the coalition from framing a member that does not belong in the coalition. A code C that ensures that an innocent authorised user is not framed by the traitors is called a frameproof code.

Definition 3.3.1. A code C over the alphabet Q is called *t-frameproof* or *narrow-sense t-frameproof*, denoted by t-FP, if for every subset D of C with $|D| \leq t$, we have that $\mathrm{desc}(D) \cap C = D$.

Definition 3.3.2. A code C over the alphabet Q is called *expanded narrow-sense t-frameproof*, denoted by t-FP*, if for every subset D of C with $|D| \leq t$, we have that $\mathrm{desc}^*(D) \cap C = D$.

Definition 3.3.3. A code C over the alphabet Q is called *wide-sense t-frameproof*, denoted by t-wFP, if for every subset D of C with $|D| \leq t$, we have that $\mathrm{wdesc}(D) \cap C = D$.

Definition 3.3.4. A code C over the alphabet Q is called *expanded wide-sense t-frameproof*, denoted by t-wFP*, if for every subset D of C with $|D| \leq t$, we have that $\mathrm{wdesc}^*(D) \cap C = D$.

Notice, that the absence of any characterisation related to the type of descendant set, implies that the narrow-sense model is used.

3.3.2 Secure Frameproof Codes

The second type of fingerprinting code is called a secure frameproof code and was first introduced in [49] by Stinson, van Trung and Wei. More precisely, they defined secure frameproof codes under the wide-sense descendant model. The situation that triggered the idea of this type of codes, was the discouraging result of Boneh and Shaw [15] proving the non existence

of deterministic wide-sense fingerprinting codes, that can identify at least one traitor. On the other hand, frameproof codes do not provide any form of traceability. Hence, secure frameproof codes were defined in order to strengthen the family of wide-sense fingerprinting codes. Let $\mathbf{x}$ be an illegal fingerprint. Since $\mathbf{x}$ could have been produced by more than one coalition, ideally we would like to identify as a traitor, the user whose fingerprint belongs to the intersection of all possible coalitions that could create $\mathbf{x}$. As this is an impossible situation (according to the result of Boneh and Shaw), Stinson, van Trung and Wei required the following property: for every pair of disjoint coalitions D_1 and D_2, their descendant sets are also disjoint. Similarly to the frameproof codes, secure frameproof codes are defined differently, depending each time on the descendant set model.

Definition 3.3.5. A code C over the alphabet Q is called *t-secure frameproof*, denoted by t-SFP, if for all distinct subsets D, D' of C such that $|D| \leq t$ and $|D'| \leq t$, we have that if $\mathrm{desc}(D) \cap \mathrm{desc}(D') \neq \emptyset$, then $D \cap D' \neq \emptyset$.

Definition 3.3.6. A code C over the alphabet Q is called *expanded narrow-sense t-secure frameproof*, denoted by t-SFP*, if for all distinct subsets D, D' of C such that $|D| \leq t$ and $|D'| \leq t$, we have that if $\mathrm{desc}^*(D) \cap \mathrm{desc}^*(D') \neq \emptyset$, then $D \cap D' \neq \emptyset$.

Definition 3.3.7. A code C over the alphabet Q is called *wide-sense t-secure frameproof*, denoted by t-wSFP, if for all distinct subsets D, D' of C such that $|D| \leq t$ and $|D'| \leq t$, we have that if $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$, then $D \cap D' \neq \emptyset$.

Definition 3.3.8. A code C over the alphabet Q is called *expanded wide-sense t-secure frameproof*, denoted by t-wSFP*, if for all distinct subsets D, D' of C such that $|D| \leq t$ and $|D'| \leq t$, we have that if $\mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D') \neq \emptyset$, then $D \cap D' \neq \emptyset$.

3.3.3 IPP Codes

Codes with the identifying parent property were first introduced by Hollmann, van Lint, Linnartz and Tolhuizen [30] (the case of two pirates, $t = 2$) and by Staddon, Stinson and Wei [47] for any set of traitors with at most t members. In contrast to the previously defined types of fingerprinting codes, IPP codes possess a strong traceability property, as they ensure the detection of at least one member of the coalition. This is achieved by identifying as a traitor the user whose fingerprint belongs to the intersection of fingerprints of all the coalitions of certain size, that could generate the illegal fingerprint. For the definition of this type of codes, it is necessary to define first the set of potential parents of a word and the descendant set of a code.

Definition 3.3.9. Let C be a code of length ℓ over the alphabet Q. For $\mathbf{x} \in Q^\ell$ define

$$\mathcal{P}_{t,C}(\mathbf{x}) = \{D \subseteq C \,:\, |D| \leq t \text{ and } \mathbf{x} \in \mathrm{desc}(D)\}$$

to be the set of all possible subsets of codewords that $\mathbf{x}$ descended from. The set $\mathcal{P}_{t,C}(\mathbf{x})$ is called the potential parent set of $\mathbf{x}$.

In the case where the wide-sense, expanded narrow-sense and expanded wide-sense descendant is being used, the corresponding potential parent sets are denoted by $\mathcal{P}^{w}_{t,C}(\cdot)$, $\mathcal{P}^{*}_{t,C}(\cdot)$ and $\mathcal{P}^{w,*}_{t,C}(\cdot)$.

Apart from the descendant set of a subset of a code, we can also define the descendant set of a code as follows:

$$\mathrm{desc}_t(C) = \bigcup_{D\subseteq C, |D|\leq t} \mathrm{desc}(D) \quad \text{(narrow-sense model)},$$

$$\mathrm{desc}^*_t(C) = \bigcup_{D\subseteq C, |D|\leq t} \mathrm{desc}^*(D) \quad \text{(expanded narrow-sense model)},$$

$$\mathrm{wdesc}_t(C) = \bigcup_{D\subseteq C, |D|\leq t} \mathrm{wdesc}(D) \quad \text{(wide-sense model)},$$

$$\mathrm{wdesc}^*_t(C) = \bigcup_{D\subseteq C, |D|\leq t} \mathrm{wdesc}^*(D) \quad \text{(expanded wide-sense model)}.$$

As expected, each of the four types of the descendant set leads to different definitions of IPP codes.

Definition 3.3.10. A code C over the alphabet Q has the *t-identifiable parent property*, if for all $\mathbf{x} \in \mathrm{desc}_t(C)$ we have that

$$\bigcap_{D \in \mathcal{P}_{t,C}(\mathbf{x})} D \neq \emptyset.$$

We denote this code by t-IPP.

Definition 3.3.11. A code C over the alphabet Q has the *expanded narrow-sense t-identifiable parent property*, if for all $\mathbf{x} \in \mathrm{desc}_t^*(C)$ we have that

$$\bigcap_{D \in \mathcal{P}_{t,C}^*(\mathbf{x})} D \neq \emptyset.$$

We denote this code by t-IPP*.

Definition 3.3.12. A code C over the alphabet Q has the *wide-sense t-identifiable parent property*, if for all $\mathbf{x} \in \mathrm{wdesc}_t(C)$ we have that

$$\bigcap_{D \in \mathcal{P}_{t,C}^w(\mathbf{x})} D \neq \emptyset.$$

We denote this code by t-wIPP.

Definition 3.3.13. A code C over the alphabet Q has the *expanded wide-sense t-identifiable parent property*, if for all $\mathbf{x} \in \mathrm{wdesc}_t^*(C)$ we have that

$$\bigcap_{D \in \mathcal{P}_{t,C}^{w,*}(\mathbf{x})} D \neq \emptyset.$$

We denote this code by t-wIPP*.

Before introducing the fourth fingerprinting code, we present a new category of codes, called *totally secure* codes that were first defined by Boneh and Shaw in [15]. The reason these codes are examined here, is because they are equivalent to IPP codes, as we will shortly prove, and hence the results of totally secure codes can be applied to IPP codes and vice versa. Like IPP

codes, totally secure codes also possess the property of identifying a traitor and in order to achieve this, the presence of a tracing algorithm is required. This algorithm is thought of as a function $A : Q^\ell \to \{1, \ldots, t\}$, which on input the illegal fingerprint $\mathbf{x} \in Q^\ell$ outputs a member of the coalition. In the original paper [15], totally secure codes were defined over $\{0, 1\}$ but here are generalised over the alphabet Q.

Definition 3.3.14 (Definition 4.1, [15])**.** Let Q be an alphabet of size q. A code C is *totally t-wSecure* code* of length ℓ, if there exists a tracing algorithm $A : Q^\ell \to \{1, \ldots, t\}$ satisfying the following condition: if a coalition D of at most t users generates a word $\mathbf{x} \in \mathrm{wdesc}^*(D)$, then $A(\mathbf{x}) \in D$.

Next, we prove that these codes are equivalent to t-wIPP* codes.

Proposition 3.3.15. *A code C is t-wIPP* if and only if C is a totally t-wSecure* code.*

Proof. First, assume that C is a t-wIPP* code of length ℓ, over the alphabet Q. Let $D_0 \subseteq C$ with $|D_0| \leq t$ and $\mathbf{x} \in \mathrm{wdesc}^*(D_0)$. In order to prove that C is totally t-wSecure*, we need to show that there exists an algorithm $A : Q^\ell \to \{1, \ldots, t\}$, such that $A(\mathbf{x}) \in D_0$. Since C is t-wIPP*, then for every $\mathbf{x} \in \mathrm{wdesc}^*_t(C)$

$$\bigcap_{D \in \mathcal{P}^{w,*}_{t,C}(\mathbf{x})} D \neq \emptyset.$$

As $D_0 \in \mathcal{P}^{w,*}_{t,C}(\mathbf{x})$, we have that given the element $\mathbf{x}$, there exists a $\mathbf{y} \in \bigcap_{D \in \mathcal{P}^{w,*}_{t,C}(\mathbf{x})} D$, which implies that $\mathbf{y} \in D_0$. In other words, there exists an algorithm $A : Q^\ell \to \{1, \ldots, t\}$, such that $A(\mathbf{x}) = \mathbf{y} \in D_0$, which means that C is totally t-wSecure*.

For the reverse direction, assume that C is a totally t-wSecure* code over Q and let $\mathbf{x_0}$ be an element of $\mathrm{wdesc}^*_t(C)$. Since C is totally t-wSecure*, there exists an algorithm $A : Q^\ell \to \{1, \ldots, t\}$ such that on input a $\mathbf{x} \in Q^\ell$ outputs a member of the coalition that produced $\mathbf{x}$. Let $\mathbf{y_0} = A(\mathbf{x_0})$ and assume for

a contradiction that the intersection of all potential sets of parents of $\mathbf{x_0}$ is empty, that is

$$\bigcap_{D \in \mathcal{P}_{t,C}^{w,*}(\mathbf{x_0})} D = \emptyset.$$

This implies that there exists a set $D \in \mathcal{P}_{t,C}^{w,*}(\mathbf{x_0})$, such that $\mathbf{y_0} \notin D$, which means that the property of totally t-wSecure code C failed for the set D. A contradiction. □

3.3.4 Traceability Codes

Traceability codes were the first type of digital fingerprinting to be introduced and were defined by Chor, Fiat and Naor [18] in order to prevent illegal redistribution of digital data. As they guarantee the identification of a traitor once the illegal fingerprint is found, traceability codes are a subset of the family of IPP codes. However, their important feature is the algorithm they provide in order to accomplish the identification of the traitor. This algorithm is deterministic and is based on the examination of the Hamming distance between codewords and words of the descendant set. First are defined the narrow-sense and wide sense traceability codes.

Definition 3.3.16. A code $C \subseteq Q^\ell$ is a *t-traceability code*, denoted t-TA, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{desc}(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d(\mathbf{x}, \mathbf{y}) < d(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d(\cdot, \cdot)$ is the Hamming distance.

Definition 3.3.17. A code $C \subseteq Q^\ell$ is a *wide-sense t-traceability code*, denoted t-wTA, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{wdesc}(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d(\mathbf{x}, \mathbf{y}) < d(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d(\cdot, \cdot)$ is the Hamming distance.

Before the definition of the expanded narrow-sense and expanded wide-sense traceability codes, we introduce the following definitions of the distance between a codeword and a word that belongs to the descendant set. This is a necessary definition, as in the expanded cases a word from the descendant set might contain the '?' symbol, which stands for the deletion of the value on that position or an unreadable mark. The existence of '?' gives rise to two different ways of defining the distance. The first, denoted by $d_1(\cdot,\cdot)$, is the known Hamming distance, whereas the second, $d_2(\cdot,\cdot)$ treats the '?' as being the same as the letter that is compared to, and thus the distance is zero.

Definition 3.3.18. For every $a \in Q$ and $b \in Q \cup \{?\}$ the distance $d_1(\cdot,\cdot)$ between a and b is defined as

$$d_1(a,b) = \begin{cases} 1, & \text{if } a \neq b \\ 1, & \text{if } b = ? \\ 0, & \text{if } a = b. \end{cases}$$

Definition 3.3.19. For every $a \in Q$ and $b \in Q \cup \{?\}$ the distance $d_2(\cdot,\cdot)$ between a and b is defined as:

$$d_2(a,b) = \begin{cases} 1, & \text{if } a \neq b \text{ and } b \in Q \\ 0, & \text{if } b = ? \\ 0, & \text{if } a = b. \end{cases}$$

It is easy to notice the relation between these two definitions of distance: for all $a \in Q$ and $b \in Q \cup \{?\}$ we have

$$d_2(a,b) = \begin{cases} d_1(a,b), & \text{if } b \neq ? \\ d_1(a,b) - 1, & \text{if } b = ?. \end{cases} \tag{3.3.1}$$

The above definitions can be easily generalised to the distance of words of length greater than 1. Let $n \geq 2$ be an integer, and $\mathbf{a} = a_1 \dots a_n \in Q^n$, $\mathbf{b} = b_1 \dots b_n \in (Q \cup \{?\})^n$ be words of length n. Then,

$$d_1(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{n} d_1(a_i, b_i)$$

and similarly

$$d_2(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} d_1(a_i, b_i).$$

The traceability code with the expanded narrow-sense descendant is next defined, under both types of distance:

Definition 3.3.20. A code $C \subseteq Q^\ell$ is an *expanded narrow-sense t-traceability code under d_1*, denoted t-TA$^*(d_1)$, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{desc}^*(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d_1(\mathbf{x}, \mathbf{y}) < d_1(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d_1(\cdot, \cdot)$ is the Hamming distance, as defined in Definition 3.3.18.

Definition 3.3.21. A code $C \subseteq Q^\ell$ is an *expanded narrow-sense t-traceability code under d_2*, denoted t-TA$^*(d_2)$, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{desc}^*(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d_2(\mathbf{x}, \mathbf{y}) < d_2(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d_2(\cdot, \cdot)$ is the distance, as defined in Definition 3.3.19.

The next proposition shows that actually the two different definitions of the distance, when applied to the expanded narrow-sense traceability code, result in equivalent codes.

Proposition 3.3.22. *A code C is t-TA$^*(d_1)$ if and only if C is t-TA$^*(d_2)$.*

Proof. It suffices to prove the claim in the case where a word $\mathbf{x}$ from the expanded narrow-sense descendant set contains a '?', because otherwise from equation (3.3.1), distances d_1 and d_2 are the same. Let r be the number of times '?' occurs in $\mathbf{x}$. First, assume that C is a t-TA$^*(d_1)$ code and let D be a subset of C of size $|D| \leq t$. Then, for all $\mathbf{y} \in D$ $d_1(\mathbf{y}, \mathbf{x}) \geq r$. Following the definition of expanded narrow-sense traceability code, for all $\mathbf{x} \in \mathrm{desc}^*(D)$

there exists $\mathbf{y_0} \in D$, such that $d_1(\mathbf{x}, \mathbf{y_0}) < d_1(\mathbf{x}, \mathbf{z})$, for all $\mathbf{z} \in C \setminus D$. The above imply that $d_1(\mathbf{x}, \mathbf{z}) \geq r$ for all $\mathbf{z} \in C \setminus D$, hence,

$$\begin{aligned} d_1(\mathbf{x}, \mathbf{y_0}) &< d_1(\mathbf{x}, \mathbf{z}) \quad \Rightarrow \\ \Rightarrow \quad d_1(\mathbf{x}, \mathbf{y_0}) - r &< d_1(\mathbf{x}, \mathbf{z}) - r \quad \Rightarrow \\ \Rightarrow \quad d_2(\mathbf{x}, \mathbf{y_0}) &< d_2(\mathbf{x}, \mathbf{z}), \end{aligned}$$

where the last inequality is derived by applying equation (3.3.1) in every position between the words $\mathbf{x}, \mathbf{y_0}$ and $\mathbf{x}, \mathbf{z}$. As the set D was chosen arbitrarily, it follows that for every $D \subseteq C$ with $|D| \leq t$ and for all $\mathbf{x} \in \mathrm{desc}^*(D)$ there exists a $\mathbf{y} \in D$, such that $d_2(\mathbf{x}, \mathbf{y}) < d_2(\mathbf{x}, \mathbf{z})$, for all $\mathbf{z} \in C \setminus D$. In other words, C is t-TA$^*(d_2)$.

The reverse direction of the claim is proved in a similar way. Assume that C is a t-TA$^*(d_2)$ code and let D be a subset of C of size $|D| \leq t$. Then, by Definition 3.3.19, for all $\mathbf{x} \in \mathrm{desc}^*(D)$ there exists $\mathbf{y_0} \in D$, such that $d_2(\mathbf{x}, \mathbf{y_0}) < d_2(\mathbf{x}, \mathbf{z})$, for all $\mathbf{z} \in C \setminus D$,. But this implies that

$$\begin{aligned} d_2(\mathbf{x}, \mathbf{y_0}) + r &< d_2(\mathbf{x}, \mathbf{z}) + r \quad \Rightarrow \\ \Rightarrow \quad d_1(\mathbf{x}, \mathbf{y_0}) &< d_1(\mathbf{x}, \mathbf{z}), \end{aligned}$$

where the last implication results from equation (3.3.1). Hence, we have proved that C is t-TA$^*(d_1)$, that is, for every D subset of C of size at most t, and for all $\mathbf{x} \in \mathrm{desc}^*(D)$ we have $d_1(\mathbf{x}, \mathbf{y_0}) < d_1(\mathbf{x}, \mathbf{z})$, for all codewords $\mathbf{z}$ that do not belong to set D. □

Finally, we present the definition of the expanded wide-sense traceability code in both types of distance.

Definition 3.3.23. A code $C \subseteq Q^\ell$ is an *expanded wide-sense t-traceability code under d_1*, denoted t-wTA$^*(d_1)$, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{wdesc}^*(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d_1(\mathbf{x}, \mathbf{y}) < d_1(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d_1(\cdot, \cdot)$ is the Hamming distance, as defined in Definition 3.3.18.

Definition 3.3.24. A code $C \subseteq Q^\ell$ is an *expanded wide-sense t-traceability code under d_2*, denoted t-wTA$^*(d_2)$, if for every $D \subseteq C$ with $|D| \leq t$ and for every $\mathbf{x} \in \mathrm{wdesc}^*(D)$, there exists at least one $\mathbf{y} \in D$ such that

$$d_1(\mathbf{x}, \mathbf{y}) < d_1(\mathbf{x}, \mathbf{z}) \quad \forall \mathbf{z} \in C \setminus D,$$

where $d_1(\cdot, \cdot)$ is the distance, as defined in Definition 3.3.19.

Similarly to the expanded narrow-sense traceability code, the t-wTA$^*(d_1)$ and t-wTA$^*(d_2)$ codes are equivalent.

Proposition 3.3.25. *A code C is t-wTA$^*(d_1)$ if and only if C is t-wTA$^*(d_2)$.*

Proof. As the only difference between the distances d_1 and d_2 is the way the positions with '?' are being counted, the fact that the detectable positions could also be any letter of the alphabet, does not complicate the proof. This is because the case where a position of a member of the expanded wide-sense set contains an arbitrary letter of Q, is treated in the same way as in the wide-sense traceability code. For this reason the proof of this proposition is identical to the proof of Proposition 3.3.22 and is omitted. □

As a consequence of Propositions 3.3.22 and 3.3.25 and also for consistency with the previous definitions of traceability codes, when referring to expanded narrow/wide-sense traceability codes, we use the Hamming distance $d_1(\cdot, \cdot)$, without specifying it, and thus use the notation t-TA* and t-wTA*.

To conclude this section, we present two remarks regarding all fingerprinting codes that have been defined.

Remark 3.3.26. By the definitions of fingerprinting codes, it is directly implied that a t-fingerprinting code C is also a t'-fingerprinting code for all $t' \leq t$. This is because for all such codes, the coalition D can have size at most t, and hence cover all cases where $|D| = t'$ for $t' \leq t$.

Remark 3.3.27. It is easy to check that any subset of a fingerprinting code is also a fingerprinting code. For example, let us examine the case of wide-sense frameproof codes. Let C be a t-wFP and C' a subset of C. Assume for a contradiction that C' is not t-wFP. This means, that there exists a subset $D \subseteq C'$ consisting of at most t codewords, for which $\mathrm{wdesc}(D) \cap C' \neq D$, or $\mathrm{wdesc}(D) \cap C' \supset D$. This is true due to the fact that $D \subseteq \mathrm{wdesc}(D)$, and thus $\mathrm{wdesc}(D) \cap C' \neq D$ implies that D is a non trivial subset of C'. As $C' \subseteq C$, we have $\mathrm{wdesc}(D) \cap C \neq D$, which leads to a contradiction to C being t-wFP.

Chapter 4

Related Work

The aim of this chapter is to describe the progress that has been made over the past years on fingerprinting codes. Results and known constructions are presented for each of the four types of fingerprinting codes, that were defined in the preceding chapter. Additionally, a section is devoted to topics closely related to these codes, for completeness.

4.1 Frameproof Codes

Recall from the previous chapter, that frameproof codes were introduced in 1995 by Boneh and Shaw [15], for the protection of innocent users. The descendant model that they used to define this type of codes is the expanded wide-sense. Furthermore, in [15] they present a construction of t-wFP* codes, based on concatenation. As an outer code they choose a binary t-wFP* code C_1 of length and size t, which contains all words that have exactly one '1'. The inner code C_2, is a (N, L, D) error-correcting code over an alphabet Q, that has minimum distance which satisfies the following expression: $D > (1 - \frac{1}{t})L$. The construction indicates that the large minimum distance constitutes a sufficient condition in order for the resulting code to be frameproof. Choosing appropriately the parameters of the component codes C_1, C_2, the result is a binary t-wFP* code of length ℓ and size $m = 2^{\ell/16t^2}$. As the construction depends on the existence of error-correcting codes with the desired characteristics, the concatenated code is not explicitly constructed. However, using expander graphs it is possible to make this construction explicit, at the expense of reducing the size of the code to

$m = 2^{\sqrt{\ell}/t}$. This construction is based on the paper by Alon, Bruck, Naor, Naor and Roth [1] on expanders.

Frameproof codes under the wide-sense model were studied by Chee [17] in his thesis, where using the same construction from expander graphs [1], in combination with explicit construction of superimposed codes of positive rate, gives the first explicit construction of binary t-wFP codes with rate bounded away from zero. Furthermore, Chee presented probabilistic constructions of binary 2-wFP code with rate $(1-o(1))\log(2/\sqrt{3})$. In the following chapter we will see that wide-sense frameproof codes are also expanded wide-sense frameproof, thus the results of Boneh and Shaw are also true for the wide-sense model as well. Hence, Chee's probabilistic results improve those of Boneh and Shaw [15] in the wide-sense model. In 1998, Stinson and Wei [50] presented a series of constructions of wide-sense frameproof codes based on different combinatorial structures. This is achieved by associating the codewords with the rows of the adjacency matrix of a set system. Since this matrix depicts the binary relation between elements and sets, the corresponding codes are defined over $\{0, 1\}$. The combinatorial structures that are used are k-designs, packing designs and perfect hash families. In particular, using packing designs Stinson and Wei presented a non explicit construction of binary 2-wFP codes, whose size $m = 2^{(\sqrt{\ell}\log\ell)/2t}$, is better than the size if the code of Boneh and Shaw, but still with rate that tends to zero. Furthermore, they introduce a method of extending existing frameproof codes in order to accommodate larger set of users. The same paper [50] also presents an upper bound on the size of t-wFP codes, an improvement of which is one of the challenges that this thesis meets, in the case where $t = 2$.

The relation between combinatorics and frameproof codes is further studied by Staddon, Stinson and Wei [47], where using cover-free and separating hash families the authors obtain an upper bound on the size of narrow-sense frameproof codes, that depends on both the length and the alphabet size.

Theorem 4.1.1 (Theorem 3.7, [47]). *For a t-FP code of length ℓ and size m over the alphabet Q of size q, the following holds:*

$$m \leq t(q^{\lceil \frac{\ell}{t} \rceil} - 1).$$

A similar problem regarding upper bounds, is considered by Blackburn in [9]. In the case where the length ℓ of the code is less than the size t of the coalition, then it is proved that the code cannot contain more than $\ell(q-1)$ codewords, where q is the alphabet size. The main focus of the paper is to examine the behaviour of the size of narrow-sense frameproof codes, when the alphabet size tends to infinity. The result presented below, is derived from the study of intersecting set systems:

Proposition 4.1.2 (Corollary 12, [9]). *Let t and ℓ be integers, and suppose that $c \geq 2$ and $\ell \geq 2$. Let $c \in \{1, \ldots, t\}$ be such that $c \equiv \ell \mod t$. Let C be a q-ary t-FP code of length ℓ. Then*

$$|C| \leq \left(\frac{\ell}{\ell - (c-1)\lceil \frac{\ell}{t} \rceil} \right) q^{\lceil \frac{\ell}{t} \rceil} + O(q^{\lceil \frac{\ell}{t} \rceil} - 1)$$

Apart from the cases where $c = 1$ and $c = t$, this bound improves upon the bound of Staddon, Stinson and Wei [47]. Additionally, the same paper [9], provides values on the size of 2-FP codes. Namely, 2-FP codes of even length ℓ consist approximately of $2q^{\lceil \ell/2 \rceil}$ codewords, while for odd ℓ the leading term becomes 1. A survey on frameproof and on fingerprinting codes in general was written by Blackburn and can be found in [8].

Staddon and Sarkar presented different constructions of narrow-sense frameproof codes in [48]. The authors observed that the union of t-FP codes of size m gives a larger code, and hence obtained t-FP codes of length ℓ and size $m' = 2^i m$ over an alphabet of size $q' = 2^i q$, where q is the size of the alphabet of the initial code. Another result of [48] is the existence of an infinite class of t-FP codes, which is shown via a recursive construction of separating hash families. Additional constructions of frameproof codes

were studied in [25], where Cohen and Encheva obtain 3-FP codes from Hadamard matrices.

In [57], Xing examines the rate of narrow-sense frameproof codes in an asymptotic manner, by fixing the alphabet size and the parameter t and letting the length tend to infinity. A lower bound on the rate in [57] is derived from algebraic curves, which can be constructed explicitly in the case where the related sequence of curves is explicit. Furthermore, this construction gives better results than the explicit constructed codes derived from error-correcting codes.

The research of frameproof codes shows that under the narrow sense model, their size is of the order of $q^{\lceil \ell/t \rceil}$. With respect to the leading coefficient of that term, the bound of Proposition 4.1.2 is the best known, except for the case $c = 1$ or $c = t$, where the best known value is given by Theorem 4.1.1. Regarding explicit construction under the narrow-sense model that produces good frameproof codes, this is given by Xing using algebraic curves, while existence results are based on good error-correcting codes of large minimum distance. The two constructions (explicit and existential) of Boneh and Shaw in [15] of expanded wide-sense frameproof codes, are based on error-correcting codes of large minimum distance and summarise the known result on frameproof codes under this model. In the case of the wide-sense model, the best explicit construction is given by Chee [17], while his probabilistic approach gives good results on binary 2-wFP codes. The best known upper bound is given by Stinson and Wei in [50]. Finally, as the following chapter shows, expanded narrow-sense frameproof codes are equivalent to narrow-sense and hence, t-FP* codes share the same results with t-FP codes.

4.2 Secure Frameproof Codes

As mentioned in the previous chapter, the idea of secure frameproof codes first appeared in the paper [49] by Stinson, van Trung and Wei, in order to enhance the frameproof codes, under the wide-sense model. In the case where the coalition is formed only by two traitors, the authors examined secure frameproof codes from a graph theory point of view and they derived the following:

Theorem 4.2.1 (Theorem 2.3, [49]). *Suppose that C is a 2-wSFP code of length ℓ and size m and suppose that $\mathbf{x}$ is an unregistered word that is produced by a coalition of size at most 2. Then, one of the following two possibilities must occur:*

1. *at least one guilty user can be identified, or*
2. *a set of three participants can be identified, two of which must be guilty.*

Apart from the graph theory, in the same paper secure frameproof codes were also studied through other combinatorial structures, such as sandwich-free families and separating systems. Furthermore, [49] presents two explicit constructions based on perfect hash families and separating hash families. The first combinatorial structure yields a code of length $\ell = 3 \cdot 7^{j+1}$ and size $m = 7^{2^j}$, for all $j \geq 0$, while for the second, the code preserves the size and reduces the length to $\ell = 9 \cdot 5^j$.

The case of narrow-sense secure frameproof codes is examined by Staddon, Stinson and Wei in [47]. Similar to [49], this paper also examines the connection between secure frameproof codes and separating hash families. As a result, the authors give an upper bound on the size of such codes:

Theorem 4.2.2 (Theorem 3.10, [47]). *For a t-SFP code of length ℓ and size m over the alphabet Q of size q, the following holds:*

$$m \leq q^{\lceil \frac{\ell}{t} \rceil} + 2t - 2. \tag{4.2.1}$$

Further research on this type of codes under the narrow-sense model, was made in [55] by Tonien and Safavi-Naini, who gave explicit constructions based on matrices defined in a specific way. They examine the case of different alphabet sizes, while for certain parameters their constructions produce exponentially large codes, compared to their length. In [36], Liu and Shen provide explicit constructions of an infinite class of separating hash families, which were derived from algebraic curves over finite fields, and were then used to obtain secure frameproof codes:

Theorem 4.2.3 (Theorem 4.6, [36]). *For any positive integers q and t, there exists an infinite class of explicitly constructed q-ary t-SFP codes of size m and length $\ell = O(\log m)$.*

Furthermore, in [51] Stinson and Zaverucha provide an upper bound on the maximum size of t-SFP codes through the existence of separating hash families.

Proposition 4.2.4 (Corollary 2.8, [51]). *If a q-ary t-SFP code of length ℓ and size m exists, then*

$$m \leq (2t^2 - 3t + 2)q^{\lceil \frac{\ell}{2t-1} \rceil} - 2t^2 + 3t - 1.$$

To conclude, under the narrow-sense model secure frameproof codes of the best asymptotic behaviour are explicitly obtained from algebraic curves, while the upper bound on the size of such codes is of the order of $q^{\lceil \frac{\ell}{2t-1} \rceil}$. Regarding the other models of descendants, only the wide-sense model was studied and explicit constructions were presented in [49] by Stinson, van Trung and Wei. As the next chapter shows, secure frameproof codes under the expanded narrow-sense model are equivalent to narrow-sense SFP codes, and hence the same results apply to t-SFP*. A similar relation connects t-wSFP* and t-wSFP codes and thus, the explicit constructions of t-wSFP can produce t-wSFP* codes, as well.

4.3 Identifying-Parent-Property Codes

Aside from the definition of frameproof codes, Boneh and Shaw also introduced totally secure codes under the expanded wide-sense model [15]. As proved in the previous chapter, this type of codes is equivalent to expanded wide-sense IPP codes. The results of their paper are rather discouraging, since they proved that there do not exist t-wSecure* (or equivalently t-wIPP*) codes for $t \geq 2$ and size $m \geq 3$. The next chapter shows that the same result holds for the IPP codes under all other descendants, apart from the narrow-sense. Hence, all the results presented in this section relate to narrow-sense IPP codes.

A first study of IPP codes, and in particular the case of two traitors, is made by Hollmann, van Lint, Linnartz and Tolhuizen in [30]. The paper provides bounds on the maximal size of 2-IPP codes of both small and arbitrary length. The small length case was also examined by Blackburn in [10], proving the following result:

Theorem 4.3.1 (Theorem 2, [10]). *Let C be a q-ary t-IPP code of length ℓ and size m. Let $u = \lceil (t/2+1)^2 \rceil$. Then, whenever $\ell < u$ we have that*

$$m \leq \frac{1}{2}u(u-1)(q-1)+1.$$

In the case of arbitrary length, Hollmann et al. [30] give explicit construction of 2-IPP codes, based on equidistant codes of length ℓ, if the distance d is odd and $\ell < (3/2)d$, if d is even. In the case where $q \geq \ell - 1$, they showed that Reed-Solomon codes with parameters $[\ell, \lceil \ell/4 \rceil, \ell - \lceil \ell/4 \rceil + 1]$ have the IPP property and consequently proved that 2-IPP codes consist of at least $q^{\lceil \ell/4 \rceil}$ codewords. For large values of q this bound was improved in the same paper by the following theorem:

Theorem 4.3.2 (Theorem 6, [30]). *Let*

$$F(\ell, q) := \max\{|C| \,:\, C \subseteq Q^\ell \text{ is 2-IPP code}, |Q| = q\}.$$

For $\ell \geq 3$ there is a constant c such that

$$F(\ell, q) \geq c\,(\frac{q}{4})^{\frac{\ell}{3}}.$$

The next chapter of the thesis shows that if a code is t-IPP, then it is also t-SFP and t-FP. Hence, results of upper bounds on the size of frameproof and secure frameproof codes could be applied to IPP codes as well. Specifically, inequality (4.2.1) given by Staddon, Stinson and Wei [47], is an upper bound for the t-IPP codes, as well. However, in the case of 2 traitors Hollman et al. [30] provide a better bound. An important result of [47] is that t-IPP codes exist under some conditions:

Proposition 4.3.3 (Corollary 2.8, [47]). *If $q \leq t$, there does not exist a q-ary t-IPP code.*

Narrow-sense IPP codes were also examined from the point of view of hypergraphs, by Barg, Cohen, Encheva, Kabatiansky and Zémor in [5]. Particularly, subsets of a t-IPP code of size at most t, are thought of as edges of a hypergraph. This approach leads to proving the following result for the asymptotic behaviour of the rate of the code:

Theorem 4.3.4 (Theorem 3.8, [5]). *Let $R_q(t) = \liminf_{\ell\to\infty} \max R(C_\ell)$, where the maximum is computed over all t-IPP codes C_ℓ of length ℓ. Let $u = \lceil (t/2+1^2) \rceil$, then*

$$R_q(t) \geq \frac{1}{u-1} \log_q \frac{(q-t)!q^u}{(q-t)!q^u - q!(q-t)^{u-t}}.$$

The above result is proved by exploiting the relation between IPP codes and partially hashing families, which are combinatorial structures introduced by the authors. Additionally, [5] examines the case of small length IPP codes and proves the existence of a sequence of linear ternary 2-IPP codes C_ℓ of length ℓ, with rate $\mathcal{R}(C_\ell) \geq (1/3)\log_3(9/7)$. In [2], Alon, Cohen, Krivelevich and Litsyn investigate this new structure and provide better results on $R_q(t)$, in the case where $q = t + 1$:

Proposition 4.3.5 (Corollary 1, [2]). *Let* $u = \lceil (t/2 + 1^2) \rceil$. *Then*

$$R_{t+1}(t) \geq \frac{t!(u-t)^{u-t}}{u^u(u-1)\ln(t+1)}.$$

The codes that provide the above results on the maximum asymptotically attainable rate are not explicit. As previously mentioned, in order for t-IPP codes to exist, it is required that $q > t$. On the other hand, when $q < t^2$, explicit constructions from error-correcting codes with large minimum distance, lead to asymptotically zero rate, since by the Plotkin bound the size of the code is of the order of q. Hence, the question posed by Barg and Kabatiansky in [6], is to explicitly construct t-IPP codes with rate bounded away from zero, provided that $t + 1 \leq q \leq t^2$. In their paper, they answer this question by constructing a sequence of t-IPP codes with asymptotically non zero rate, based on concatenation of an IPP and a linear code. Choosing the component codes carefully (for example by taking a Reed-Solomon code as the linear code) and the fact that the decoding uses the Guruswami-Sudan algorithm, the identification is efficient under certain conditions.

New bounds on the maximum size of narrow-sense t-IPP of length ℓ, size m and alphabet size q, are proved by Alon and Stav [3]. The upper bound in the following result is an improvement on a bound given in [10] by Blackburn.

Theorem 4.3.6 (Theorem 2.1, [3]). *There exist two functions* $c_1(t)$ *and* $c_2(t)$, *such that for every* ℓ, q

$$(c_1(t)q)^{\frac{\ell}{s(t)}} < m < c_2(t)q^{\lceil \frac{\ell}{s(t)} \rceil},$$

where

$$s(t) = \begin{cases} \frac{t^2}{4} + t, & \text{if } t \text{ is even,} \\ \frac{t^2}{4} + t - \frac{1}{4}, & \text{if } t \text{ is odd.} \end{cases}$$

The above results show that in the case of small length, the maximum size of t-IPP codes is of the order of the alphabet size q. Regarding arbitrary

length, the results can be interpreted in two ways. The first one would be to fix the length ℓ and the parameter t of the IPP codes and let q tend to infinity. Under this hypothesis and using the inequalities of the last theorem, we can see that the maximum attainable rate is approximately bounded below from $1/s(t)$ and $(1/\ell)\lceil \ell/s(t) \rceil$ from above. If instead the alphabet size q is fixed and the length ℓ grows, then the result is given by Theorem 4.3.4.

4.4 Traceability Codes

The first note on traceability codes and their construction was made in 1994 by Chor, Fiat and Naor [18]. The paper begins with a discussion on the problem of tracing traitors and continues by giving an application of fingerprinting in the broadcast setting, under the narrow-sense model. The t- and (t, p)-traceability schemes are defined, where the latter contains a probabilistic factor. Additionally, the authors prove the existence of a t-traceability scheme and provide a sufficient condition for traceability. In particular, they translated their construction into the language of coding theory and thus, the sufficient condition becomes an expression of the minimum distance of a code: find a code of size m, length $\ell = 4t^2 \log m$ and minimum distance d, over an alphabet of size $q = 2t^2$, such that $d > (1 - \frac{1}{t^2})\ell$.

Using the same model of descendants, Stinson and Wei [50] explore binary traceability codes and their relation to set systems. A series of existence results on t-TA codes is presented, based on k-designs and packing designs. In particular, packing designs give traceability codes of size approximately $2^{\ell \log \ell / t^2}$, which is better than the size of the code given in [18]. However, the alphabet size of the latter is much smaller. In [50], the authors also give an upper bound on the maximum size of t-TA codes:

Theorem 4.4.1 (Theorem 5.5, [50]). *If a q-ary t-TA code of length ℓ and size m*

exists, then the following bound holds

$$m \leq \frac{\binom{q}{\lceil \frac{\ell}{t} \rceil}}{\binom{\ell-1}{\lceil \frac{\ell}{t} \rceil - 1}}.$$

As in the case of frameproof codes, the same paper presents a method of enlarging the size of these codes, in order to cover bigger set of users. Binary, narrow-sense traceability codes were also examined in [47] by Staddon, Stinson and Wei. Upper bounds on the size of t-TA codes were indirectly derived from t-SFP codes, by exploiting the fact that traceability codes are also secure-frameproof codes. Thus, according to inequality (4.2.1) a t-TA code has at most $q^{\lceil \ell/t \rceil} + 2t - 2$ codewords. Using the result of Chor, Fiat and Naor [18] regarding the connection between traceability codes and codes of large minimum distance, Staddon, Stinson and Wei [47] presented a q-ary t-TA code over an alphabet of size a prime power, length $\ell \leq q + 1$, which contains $q^{\lceil \ell/t^2 \rceil}$ codewords. The construction is based on the existence of Reed-Solomon codes of certain parameters.

The flavour of coding theory that accompanies traceability codes, introduces efficient tracing methods that relate to error-correcting codes. One of these methods appears in a paper by Silverberg, Staddon and Walker [45], where the authors use list decoding on linear t-TA codes and narrow-sense traceability codes based on Reed-Solomon codes of large minimum distance.

In [12], Blackburn, Etzion and Ng proved that in the case of two traitors, the size of 2-TA codes of length ℓ is at most $cq^{\lceil t/4 \rceil}$, where c is a parameter depending on ℓ. As mentioned in [47], when $\ell \leq q + 1$ there exists a Reed-Solomon code with at most $q^{\lceil t/4 \rceil}$. Hence, the fact that a Reed-Solomon code is 2-TA with size of the same order as the bound given by [12], shows that error-correcting codes of large minimum distance result in good 2-TA codes. Moreover [12] proves the following result, which shows the existence of constant rate traceability codes of specific parameters:

Theorem 4.4.2 (Theorem 2, [12]). *Let t and q be integers with $t \geq 2$. When*

$$t^2 - \lceil \frac{t}{2} \rceil + 1 \leq q \quad \text{or when } t = 2 \text{ and } q = 3,$$

there exists a constant $R > 0$ (that depends on q and t) and a sequence of t-TA codes $C_1, C_2, \ldots$ over an alphabet of size q, such that C_ℓ has length ℓ and $|C_\ell| \sim q^{R\ell}$ as $\ell \to \infty$.

From the above presentation of results we conclude that the upper bound on the size of traceability codes under the narrow-sense model, is of the order of $q^{\lceil \ell/t \rceil}$, though several constructions yield traceability codes of size approximately $q^{\lceil \ell/t^2 \rceil}$. In the case of two traitors, it is shown that the upper bound is $O(q^{\lceil \ell/4 \rceil})$ and can be attained using error-correcting codes. Regarding the other three models of the descendant set, the following chapter shows that t-TA*, t-wTA and t-wTA* are equivalent to t-wIPP* codes, which according to the result of Boneh and Shaw [15] do not exist for $t \geq 2$ and and size $m \geq 3$.

4.5 Beyond the Main Types of Fingerprinting Codes

To conclude the chapter, this section describes in brief and for the purpose of completeness two additional topics related to digital fingerprinting. The first concerns *t-secure ε-error* codes, while the second refers to digital fingerprinting in the public key setting.

4.5.1 Secure ε-Error Codes

As mentioned in a previous section (Section 4.3), the results of IPP codes under the expanded wide-sense models are quite discouraging, as in [15] Boneh and Shaw proved that t-wIPP* codes exist only when $t \leq 1$ or $m \leq 2$. For this reason, they defined a new type of fingerprinting code which contain some randomness $r \in \{0,1\}^*$ in the choices made by the distributor, when he associates fingerprints with users. By keeping secret the way fin-

gerprints are distributed, they construct a scheme which identify a traitor with high probability. The randomised fingerprinting scheme is now called t-secure ε-error, where the parameter ε refers to the probability of misidentification of a traitor.

Definition 4.5.1 (Definition 4.2, [15])**.** A fingerprinting scheme C_r is t-secure with ε-error if there exists a tracing algorithm A satisfying the following condition: if a coalition D of at most t users generates a word $\mathbf{x}$ then

$$Pr[A(\mathbf{x}) \in D] > 1 - \varepsilon,$$

where the probability is taken over the random bits r and the random choices made by the coalition.

Randomized fingerprinting schemes were also studied in [4], by Barg, Blakley and Kabatiansky. Using concatenation of codes, they obtain a binary totally t-secure ε-error code with rate bounded away from zero. The authors choose for the inner code a binary t-SFP code and for the outer, an extended q-ary Reed-Solomon code. The paper also examines the case of two traitors, and based on the aforementioned construction with a 2-SFP as inner code, they obtain a 2-secure code with the following property: either one traitor is identified with probability 1, or both with probability $1 - \varepsilon$. More 2-secure ε-error codes are constructed by Tô, Safavi-Naini and Wang in [54], using the method of concatenation, as well. The inner code is a binary 2-secure ε'-error code, while there are two choices for the outer structure: t-wTA* code or perfect hash families. The main result of this paper is that the probability of the error ε' is very small and in the case where the illegal fingerprint contains at least one '?', the identification is correct.

Another way of addressing the problem of tracing traitors, is to use public key schemes instead of combinatorial. The motivation behind this approach, is that the symmetric model is based on the assumption of the honesty of the distributor. Since the distributor holds all the information

on the fingerprints, it is possible to frame an innocent user. This scenario was first considered by Pfitzmann and Schunter in [41], who presented constructions of asymmetric tracing schemes by combining known symmetric fingerprinting schemes with cryptographic primitives that are provably secure. Further investigation on the asymmetric model was made by Pfitzmann [40], Kurosawa and Desmedt [34] and Pfitzmann and Waidner [42], who constructed schemes that address the problem of collusion of users. Boneh and Franklin in [14] describe a more efficient tracing scheme, compared to the previous ones, based on Reed-Solomon codes and under the assumption that the decisional Diffie-Hellman problem is hard. In [38] Naccache, Shamir and Stern introduce a technique for copyrighting functions, algorithms and programs, that does not depend on the marking assumption. As a result, either a traitor is traced or the data is extremely altered. Kurosawa and Yoshida [35] use linear codes to obtain public key tracing schemes and in [33] Kiayias and Yung present two tracing schemes with constant transmission rate (that is, the growth of the size of keys and ciphertexts in relation to the plaintext size). An improvement to the transmission rate of the scheme proposed by Kiayias and Yung, was given in [16] by Chabanne, Phan and Pointcheval. The same paper introduces the notion of public traceability, which also allows the users, apart from the distributor, to perform tracing.

This area, combining traitor tracing schemes with public key cryptography, possesses a large collection of tracing schemes and continues to produce new results in order to cover the needs of protecting intellectual rights.

Chapter 5

Relations Between Fingerprinting Codes

This chapter presents a series of results with regard to the connections of all sixteen types of fingerprinting codes, derived by combining the four security notions with the four adversary models. Previous research has covered the relations between traceability and IPP codes, and secure frameproof and frameproof codes, under the narrow-sense descendant set. Proposition 5.1.2 bridges the gap by examining the relation between narrow-sense IPP and secure frameproof codes. Regarding the remaining relations, these are presented by Propositions 5.2.2, 5.2.3, 5.2.4, 5.2.13, 5.3.1, 5.3.2, 5.3.3, 5.3.4, 5.3.5, 5.3.6, 5.3.7 and 5.3.8. The results of these propositions are new, though they require only minor modifications of previously known results. Additionally, based on a result by Boneh and Shaw [15], Theorems 5.2.8, 5.2.9, 5.2.10, 5.2.11 and 5.2.12 present further original results, which show that IPP and traceability codes under the wide-sense and the expanded wide-sense model are equivalent.

5.1 The Narrow-Sense Model

This section investigates the relations between different fingerprinting codes, which use the same type of descendant set. The first relation involves traceability and IPP codes, and was proved by Staddon, Stinson and Wei in [47].

Proposition 5.1.1 (Lemma 1.3, [47]). *A t-TA code is a t-IPP code.*

Proof. Let C be a t-TA code and D be a subset of C with $|D| \leq t$. Let $\mathbf{x_0} \in \mathrm{desc}(D)$ and $\mathbf{y} \in D$, such that $d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z})$ for all $\mathbf{z} \in D$. Combining

this with the property of t-TA codes, we obtain:

$$d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z}), \text{ for all } \mathbf{z} \in C. \tag{5.1.1}$$

Assume there exists a set $D' \subseteq C$ of size at most t, for which $\mathbf{x_0} \in \mathrm{desc}D'$ and $\mathbf{y} \notin D'$. Then, as C is t-TA, there must exist a codeword $\mathbf{w} \in D'$, such that $d(\mathbf{x_0}, \mathbf{w}) < d(\mathbf{x_0}, \mathbf{y})$. But this contradicts equation (5.1.1). Hence, for every subset D of C with $|D| \leq t$ and for all $\mathbf{x} \in \mathrm{desc}(D)$, there exists a $\mathbf{y} \in C$ such that $\mathbf{y} \in \bigcap_{D \in \mathcal{P}_{t,C}(\mathbf{x})} D$, in other words, C is t-IPP. □

Next, an example is presented demonstrating that the above relation is one way and that a t-IPP code is not necessarily t-TA.

Example 5.1.1. The code $C = \{000, 011, 102, 220\}$, generated from the construction introduced by Hollmann et al. in [30], is a 2-IPP code. Let $D = \{011, 102\}$ and $\mathbf{x} = 001$ be a word which belongs to $\mathrm{desc}(D)$. Then, there exists a $\mathbf{z} = 000 \in C \setminus D$, such that $d(\mathbf{x}, \mathbf{y}) \geq d(\mathbf{x}, \mathbf{z}) = 1$ for all $\mathbf{y} \in D$, which contradicts the definition of traceability codes. Hence, C is 2-IPP but not 2-TA.

A similar relation connects IPP with secure frameproof codes.

Proposition 5.1.2. *A t-IPP code is a t-SFP code.*

Proof. Let C be a t-IPP code and D, D' subsets of C with $|D| \leq t$ and $|D'| \leq t$. Additionally, let $\mathbf{x}$ be a word in the intersection of $\mathrm{desc}(D)$ and $\mathrm{desc}(D')$, and thus we have $D, D' \in \mathcal{P}_{t,C}(\mathbf{x})$. Then, as C is t-IPP, there exists a $\mathbf{y} \in \bigcap_{D_i \in \mathcal{P}_{t,C}(\mathbf{x})} D_i$. Notice, that

$$\bigcap_{D_i \in \mathcal{P}_{t,C}(\mathbf{x})} D_i \subseteq D \cap D'$$

and consequently $\mathbf{y} \in D \cap D'$. Hence, for all distinct D, D' subsets of C of size at most t when $\mathrm{desc}(D) \cap \mathrm{desc}(D') \neq \emptyset$, then $D \cap D' \neq \emptyset$, which proves that C is t-SFP. □

Example 5.1.2. The code $C = \{1002, 1201, 2001, 2212\}$ is a 2-SFP code. For the word $\mathbf{x} = 2002$ in the set $\mathrm{desc}_2(C)$ the potential parent sets of size 2 are:

$D_1 = \{1002, 2001\}$

$D_2 = \{1002, 2212\}$

$D_3 = \{2001, 2212\}$,

which do not intersect, and hence C is not 2-IPP.

Finally, we present the connection between t-SFP and t-FP codes, which was examined in [49] by Stinson et al.

Proposition 5.1.3 (Theorem 2.2, [49])**.** *A t-SFP code is a t-FP code.*

Proof. Let C be a t-SFP code. Assume for a contradiction, that C is not t-FP. Then, there exists a subset D of C with $|D| \leq t$, such that $\mathrm{desc}(D) \cap C \neq D$. Since $D \subseteq C$, we have that $D \subseteq \mathrm{desc}(D) \cap C$, which implies that here exists a word $\mathbf{y}$ such that $\mathbf{y} \in \mathrm{desc}(D)$ and $\mathbf{y} \in C$, but $\mathbf{y} \notin D$. Let $\{\mathbf{y}\} = D'$, then D, D' are subsets of C of size at most t and have empty intersection, $D \cap D' = \emptyset$. As $\{\mathbf{y}\} \in \mathrm{desc}(D)$ and $\{\mathbf{y}\} \in \mathrm{desc}(D')$, we have that $\mathrm{desc}(D) \cap \mathrm{desc}(D') \neq \emptyset$, which contradicts the fact that C is t-SFP. Hence, C is indeed t-FP. □

Example 5.1.3. Let us examine the 2-FP code $C = \{10110, 10201, 11000, 00100\}$. For the disjoint subsets $D = \{10110, 10201\}$ and $D' = \{11000, 00100\}$ of C, there exists a word $\mathbf{x} = 10100$ which belongs to both $\mathrm{desc}(D)$ and $\mathrm{desc}(D')$. Hence, C cannot be 2-SFP, as $D \cap D' = \emptyset$ and the definition of the t-SFP codes is violated.

So far, the relations amongst the fingerprinting codes are summarised in Figure 5.1.

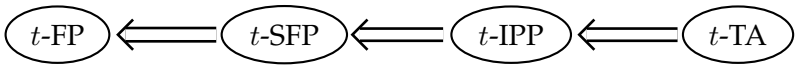


Figure 5.1: Relations of fingerprinting codes under the narrow-sense model.

5.2 Wide-Sense and Expanded Narrow/Wide-Sense Models on Traceability and IPP Codes

Before proceeding with the examination of the remaining relations of fingerprinting codes, we make the following remark regarding the size of the codes.

Remark 5.2.1. Let us examine the cases where C is a fingerprinting code of size 1 or 2. Let $\mathbf{x}$ be an illegal fingerprint under one of the four descendant models. When $|C| = |\{\mathbf{y}\}| = 1$, then the illegal fingerprint $\mathbf{x}$ coincides with the codeword $\mathbf{y}$, as the traitor knows nothing more than the letters of his own fingerprint. In the case where $|C| = |\{\mathbf{y_1}, \mathbf{y_2}\}| = 2$ and $\mathbf{x} \in C$, then $\mathbf{x} = \mathbf{y_1}$ or $\mathbf{x} = \mathbf{y_2}$, and one traitor is directly identified. When $|C| = |\{\mathbf{y_1}, \mathbf{y_2}\}| = 2$ and $\mathbf{x} \notin C$, then both codewords must correspond to traitors, as the only way to produce an illegal fingerprint that does not belong to C, is by combining their fingerprints. As in both cases, $|C| = 1$ and $|C| = 2$, the identification of a traitor is direct, we call these codes *trivial*.

Next, we examine the connection between traceability and IPP codes using the expanded narrow-sense, wide-sense and expanded wide-sense models of descendant. Similar to the narrow-sense model, traceability codes imply IPP codes under the remaining descendant models, as well. As the proofs are identical to the proof of Proposition 5.1.1 and the only difference is the use of the descendant set, they can be disregarded and are presented here only for completeness.

Proposition 5.2.2. *A t-TA* code is a t-IPP* code.*

Proof. Let C be a t-TA* code and $D \subseteq C$ of size at most t. Let $\mathbf{x_0} \in \mathrm{desc}^*(D)$ and $\mathbf{y}$ be the closest element to $\mathbf{x}$ in D, that is, $d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z})$ for all $\mathbf{z} \in D$. Since C is t-TA*, we have

$$d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z}), \text{ for all } \mathbf{z} \in C. \tag{5.2.1}$$

Assume for a contradiction that there exists $D' \subseteq C$ with $|D'| \leq t$, such that $\mathbf{x_0} \in \mathrm{desc}^* D'$ and $\mathbf{y} \notin D'$. Then, by the definition of t-TA* code there exists a codeword $\mathbf{w} \in D'$ such that $d(\mathbf{x_0}, \mathbf{w}) < d(\mathbf{x_0}, \mathbf{y})$, a contradiction to (5.2.1). Hence, for every subset D of C with $|D| \leq t$ and for all $\mathbf{x} \in \mathrm{desc}^*(D)$, there exists a $\mathbf{y} \in C$ such that $\mathbf{y} \in \bigcap D \in \mathcal{P}^*_{t,C}(\mathbf{x})D$, in other words, C is t-IPP*. □

Proposition 5.2.3. *A t-wTA code is a t-wIPP code.*

Proof. Let C be a t-wTA code and $D \subseteq C$ of size at most t. Let $\mathbf{x_0} \in \mathrm{wdesc}(D)$ and $\mathbf{y}$ be the closest element to $\mathbf{x}$ in D, that is, $d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z})$ for all $\mathbf{z} \in D$. Since C is t-wTA, we have

$$d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z}), \text{ for all } \mathbf{z} \in C. \tag{5.2.2}$$

Assume for a contradiction that there exists $D' \subseteq C$ with $|D'| \leq t$, such that $\mathbf{x_0} \in \mathrm{wdesc} D'$ and $\mathbf{y} \notin D'$. Then, by the definition of t-wTA code there exists a codeword $\mathbf{w} \in D'$ such that $d(\mathbf{x_0}, \mathbf{w}) < d(\mathbf{x_0}, \mathbf{y})$, a contradiction to (5.2.2). Hence, for every subset D of C with $|D| \leq t$ and for all $\mathbf{x} \in \mathrm{wdesc}(D)$, there exists a $\mathbf{y} \in C$ such that $\mathbf{y} \in \bigcap D \in \mathcal{P}^w_{t,C}(\mathbf{x})D$, in other words, C is t-wIPP. □

Proposition 5.2.4. *A t-wTA* code is a t-wIPP* code.*

Proof. Let C be a t-wTA* code and $D \subseteq C$ of size at most t. Let $\mathbf{x_0} \in \mathrm{wdesc}^*(D)$ and $\mathbf{y}$ be the closest element to $\mathbf{x}$ in D, that is, $d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z})$ for all $\mathbf{z} \in D$. Since C is t-wTA*, we have

$$d(\mathbf{x_0}, \mathbf{y}) \leq d(\mathbf{x_0}, \mathbf{z}), \text{ for all } \mathbf{z} \in C. \tag{5.2.3}$$

Assume for a contradiction that there exists $D' \subseteq C$ with $|D'| \leq t$, such that $\mathbf{x_0} \in \mathrm{wdesc}^* D'$ and $\mathbf{y} \notin D'$. Then, by the definition of t-wTA* code there exists a codeword $\mathbf{w} \in D'$ such that $d(\mathbf{x_0}, \mathbf{w}) < d(\mathbf{x_0}, \mathbf{y})$, a contradiction to (5.2.3). Hence, for every subset D of C with $|D| \leq t$ and for all $\mathbf{x} \in \mathrm{wdesc}^*(D)$, there exists a $\mathbf{y} \in C$ such that $\mathbf{y} \in \bigcap D \in \mathcal{P}^{w,*}_{t,C}(\mathbf{x})D$, in other words, C is t-wIPP*. □

Now, let us recall for a moment totally t-secure codes, defined in Chapter 3, and present a result by Boneh and Shaw. We have already proved that totally t-wSecure* codes are equivalent to t-wIPP*, hence, the result on totally secure codes holds for IPP codes, as well.

Theorem 5.2.5 (Theorem 4.2, [15]). *There are no totally t-wSecure* codes C for $t \geq 2$ and size $|C| \geq 3$.*

The proof of the above theorem is based on the next lemma.

Lemma 5.2.6 (Lemma 4.1, [15]). *If C is a totally t-wSecure* code, then*

$$D_1 \cap D_2 \cap \ldots \cap D_r = \emptyset \quad \Rightarrow \quad \mathrm{wdesc}^*(D_1) \cap \mathrm{wdesc}^*(D_2) \cap \ldots \cap \mathrm{wdesc}^*(D_r) = \emptyset,$$

for all subsets $D_1, D_2, \ldots, D_r$ of C of size at most t.

Proof. Let C be a totally t-wSecure* code and $D_1, D_2, \ldots, D_r$ subsets of C of size at most t with their intersection being the empty set. Assume for a contradiction, that $\mathrm{wdesc}^*(D_1) \cap \mathrm{wdesc}^*(D_2) \cap \ldots \cap \mathrm{wdesc}^*(D_r) = \mathbf{x} \neq \emptyset$. This implies that all sets $D_1, D_2, \ldots, D_r$ are suspect, as they correspond to potential traitors. Since these sets do not intersect, it is not possible to determine which element is associated with the traitor, a contradiction to the fact that there exists an algorithm that identifies a traitor. Hence, when a code is totally t-wSecure* and $D_1 \cap D_2 \cap \ldots \cap D_r = \emptyset$, then the sets $\mathrm{wdesc}^*(D_1)$, $\mathrm{wdesc}^*(D_2)$, ..., $\mathrm{wdesc}^*(D_r)$ must also have empty intersection. □

Proof of Theorem 5.2.5. It is sufficient to prove the claim for the totally 2-wSecure* codes, since the non existence of totally 2-wSecure* codes of size more that 2, implies the non existence of larger totally t-wSecure* codes with $t > 2$.

Let C be a code of length ℓ and size m over an alphabet Q and $\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3$ be distinct codewords. Define the majority word $\mathbf{x} = MAJ(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3)$ by

$$x_i = \begin{cases} y_i^1, & \text{if } y_i^1 = y_i^2 \text{ or } y_i^1 = y_i^3 \\ y_i^2, & \text{if } y_i^2 = y_i^3 \\ ?, & \text{otherwise} \end{cases}$$

Suppose that the coalition D consists of the codewords $D = \{\mathbf{y}^1, \mathbf{y}^2\}$. Then one of the words in the set $\mathrm{wdesc}^*(D)$ is the majority word $\mathbf{x}$. Furthermore, this word belongs also to $\mathrm{wdesc}^*(\{\mathbf{y}^1, \mathbf{y}^3\})$ and to $\mathrm{wdesc}^*(\{\mathbf{y}^2, \mathbf{y}^3\})$, which means that

$$\mathbf{x} \subseteq \mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(\{\mathbf{y}^1, \mathbf{y}^3\}) \cap \mathrm{wdesc}^*(\{\mathbf{y}^2, \mathbf{y}^3\}).$$

However, the intersection of the coalitions $D, \{\mathbf{y}^1, \mathbf{y}^3\}$ and $\{\mathbf{y}^2, \mathbf{y}^3\}$ is the empty set and thus by Lemma 5.2.6, C is not totally 2-wSecure*. □

Using the above theorem together with Proposition 3.3.15, we derive the following result on IPP codes.

Theorem 5.2.7. *There are no non trivial t-wIPP* codes.*

Combining the previous theorem with the fact that any code C that is t-wTA* is also t-wIPP*, we obtain that t-wTA* codes are also trivial.

Theorem 5.2.8. *There are no non trivial t-wTA* codes.*

Proof. Assume for a contradiction, that there exists a non trivial t-TA* code C. Then by Proposition 5.2.4 C is also a non trivial t-wIPP* code, which contradicts Theorem 5.2.7. □

Next, we prove similar results regarding the wide-sense traceability and IPP codes.

Theorem 5.2.9. *There are no non trivial t-wTA codes.*

Proof. By Remark 3.3.26, it suffices to prove the claim for the 2-wTA codes. Let C be a 2-wTA code of length ℓ and size $m \geq 3$. In order to reach a contradiction, is enough to find a subset D of C of size 2, for which the following hold: there exists at least one $\mathbf{x} \in \mathrm{wdesc}(D)$, such that for all $\mathbf{y} \in D$

$$d(\mathbf{x}, \mathbf{z}) < d(\mathbf{x}, \mathbf{y}), \quad \text{for some } \mathbf{z} \in C \setminus D. \tag{5.2.4}$$

Note that it is sufficient to prove the non-existence of a 2-wTA code of size $m = 3$. This is because any code C with $m \geq 3$ has a subset of size 3 from which we can derive the contradiction. So let $C = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$ be a 2-wTA code of size $m = 3$. Before we continue, it is necessary to introduce some notation. We denote by $I_{j,k}$ the set of positions where the words $\mathbf{y^j}$ and $\mathbf{y^k}$ agree and for every combination of words we use the letter a to denote the corresponding cardinality of the sets:

$$I_{123} = \{i \in \{1, \ldots, n\} : \mathbf{y}_i^1 = \mathbf{y}_i^2 = \mathbf{y}_i^3\}, \quad a_1 = |I_{123}|$$

$$I_{12} = \{i \in \{1, \ldots, n\} : \mathbf{y}_i^1 = \mathbf{y}_i^2 \text{ and } \mathbf{y}_i^1 \neq \mathbf{y}_i^3\}, \quad a_2 = |I_{12}|$$

$$I_{13} = \{i \in \{1, \ldots, n\} : \mathbf{y}_i^1 = \mathbf{y}_i^3 \text{ and } \mathbf{y}_i^1 \neq \mathbf{y}_i^2\}, \quad a_3 = |I_{13}|$$

$$I_{23} = \{i \in \{1, \ldots, n\} : \mathbf{y}_i^2 = \mathbf{y}_i^3 \text{ and } \mathbf{y}_i^2 \neq \mathbf{y}_i^1\}, \quad a_4 = |I_{23}|$$

Also, let I_0 denote the set of positions where all codewords disagree:

$$I_0 = \{i \in \{1, \ldots, n\} : \mathbf{y}_i^1 \neq \mathbf{y}_i^2, \mathbf{y}_i^2 \neq \mathbf{y}_i^3, \mathbf{y}_i^1 \neq \mathbf{y}_i^3\}, \quad a_5 = |I_0|$$

Note that a_1, a_2, a_3, a_4, a_5 must add up to ℓ, which is the length of the code. Without loss of generality, suppose that $a_2 \leq \min\{a_3, a_4\}$. Let $D = \{\mathbf{y}^1, \mathbf{y}^2\}$. We generate the descendant $\mathbf{x} \in \mathrm{wdesc}(D)$ as follows:

$$x_i = \begin{cases} y_i^1, & \text{if } y_i^1 = y_i^2 \\ y_i^3, & \text{otherwise.} \end{cases}$$

Next, we calculate the distances between the codewords and the descendant $\mathbf{x}$:

$$d(\mathbf{x}, \mathbf{y}^1) = \ell - |I_{12}| - |I_{13}| - |I_{123}| = a_4 + a_5$$

$$d(\mathbf{x}, \mathbf{y}^2) = \ell - |I_{12}| - |I_{23}| - |I_{123}| = a_3 + a_5$$

$$d(\mathbf{x}, \mathbf{y}^3) = |I_{12}| = a_2$$

Since $a_2 \leq \min\{a_4, a_3\}$, we have that $\mathbf{y}^3$, which is not a member of the coalition D, is closer to the descendant $\mathbf{x}$, a contradiction to the definition of a 2-wTA code.

Now assume that C is a trivial code. We prove that C is 2-wTA. If C has only one codeword, say $C = \{\mathbf{y}\}$, then $\mathbf{y}$ belongs to the set D, as well as to the wide-sense descendant set of D. In this case, the properties of a 2-wTA code are trivially true. Let $C = \{\mathbf{y}^1, \mathbf{y}^2\}$ with $\mathbf{y}^1 \neq \mathbf{y}^2$. If D consists of only one codeword, say $\mathbf{y^1}$, then $\mathbf{y^1} \in \mathrm{wdesc}(D)$ and hence C is 2-wTA since $d(\mathbf{y^1}, \mathbf{y^1}) < d(\mathbf{y}^1, \mathbf{y^2})$. If D contains two elements, we have the situation where $D = C$, and thus the identification of at least one traitor is straightforward. □

Another way of proving the same result as Theorem 5.2.9 is through t-wIPP codes. The idea is to exploit the relation between t-wTA and t-wIPP codes, along with the following theorem stating that all t-wIPP codes are trivial.

Theorem 5.2.10. *There are no non trivial t-wIPP codes.*

Proof. The proof is a modified version of the proof of the Theorem 5.2.5, on totally secure codes. Again by Remark 3.3.26 we only need to prove the claim for the case of 2-wIPP codes. Let C be a non trivial 2-wIPP code over the alphabet Q, specifically let $C = \{\mathbf{y^1}, \mathbf{y^2}, \mathbf{y^3}\}$. Take the element $\mathbf{x} \in \mathrm{wdesc}_2(C)$ to be the majority word

$$x_i = \begin{cases} y_i^1, & \text{if } y_i^1 = y_i^2 \text{ or } y_i^1 = y_i^3 \\ y_i^2, & \text{if } y_i^2 = y_i^3 \\ \alpha, & \text{otherwise,} \end{cases}$$

where α is an arbitrary letter of the alphabet Q. Since $\mathbf{x}$ has been produced with the wide-sense model, we have that $\mathbf{x} \in \mathrm{wdesc}_2(C)$. By the definition of 2-wIPP codes, this implies that the intersection of all potential parent sets of $\mathbf{x}$ must be non empty. For the majority word $\mathbf{x}$, the potential parent sets

are all the possible 2-subsets of C:

$D_1 = \{\mathbf{y^1}, \mathbf{y^2}\} \in \mathcal{P}^w_{2,C}(\mathbf{x})$,

$D_2 = \{\mathbf{y^1}, \mathbf{y^3}\} \in \mathcal{P}^w_{2,C}(\mathbf{x})$,

$D_3 = \{\mathbf{y^2}, \mathbf{y^3}\} \in \mathcal{P}^w_{2,C}(\mathbf{x})$,

but since $D_1 \cap D_2 \cap D_3 = \emptyset$, we reached a contradiction. Hence, all 2-wIPP codes are trivial. □

The same result holds for the t-IPP* codes as well, and as the proof follows the proof of Theorem 5.2.10, we include it only for completeness.

Theorem 5.2.11. *There are no non trivial t-IPP* codes.*

Proof. By Remark 3.3.26 we only need to prove the claim for the case of 2-IPP* codes. Let C be a non trivial 2-IPP* code over the alphabet Q, specifically let $C = \{\mathbf{y^1}, \mathbf{y^2}, \mathbf{y^3}\}$. Take the element $\mathbf{x} \in \mathrm{desc}^*_2(C)$ to be the majority word

$$x_i = \begin{cases} y^1_i, & \text{if } y^1_i = y^2_i \text{ or } y^1_i = y^3_i \\ y^2_i, & \text{if } y^2_i = y^3_i \\ ?, & \text{otherwise,} \end{cases}$$

Since $\mathbf{x}$ has been produced with the expanded narrow-sense model, we have that $\mathbf{x} \in \mathrm{desc}^*_2(C)$. By the definition of 2-IPP codes, this implies that the intersection of all potential parent sets of $\mathbf{x}$ must be non empty. For the majority word $\mathbf{x}$, the potential parent sets are all the possible 2-subsets of C:

$D_1 = \{\mathbf{y^1}, \mathbf{y^2}\} \in \mathcal{P}^*_{2,C}(\mathbf{x})$,

$D_2 = \{\mathbf{y^1}, \mathbf{y^3}\} \in \mathcal{P}^*_{2,C}(\mathbf{x})$,

$D_3 = \{\mathbf{y^2}, \mathbf{y^3}\} \in \mathcal{P}^*_{2,C}(\mathbf{x})$,

but since $D_1 \cap D_2 \cap D_3 = \emptyset$, we have a contradiction. Hence, all 2-IPP* codes are trivial. □

Once more, we exploit the relation between traceability codes and IPP codes, in order to obtain a similar result for the t-TA* codes.

Theorem 5.2.12. *There are no non trivial t-TA* codes.*

Proof. By Proposition 5.2.2, the existence of non trivial t-TA* codes imply the existence of non trivial t-IPP* code, which contradicts Theorem 5.2.11. □

The aforementioned theorems and propositions, regarding traceability and IPP codes, lead to the conclusion that these codes are indeed equivalent, since they are all trivial (Figure 5.2).

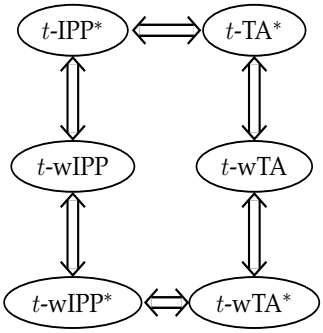


Figure 5.2: Relations of traceability and IPP codes under the expanded narrow-sense, wide-sense and expanded wide-sense model.

To connect these results with the fingerprinting codes under the narrow-sense model, it suffices to prove an implication or an equivalence between t-TA and t-wTA codes.

Proposition 5.2.13. *A t-wTA code is a t-TA code.*

Proof. Let C be a t-wTA code and $D \subseteq C$ with $|D| \leq t$. Let $\mathbf{x} \in \mathrm{desc}(D)$ and $\mathbf{y} \in C$ be the closest codeword to $\mathbf{x}$ than any other $\mathbf{z} \in C$. In order to prove that C is t-TA, we need to show that $\mathbf{y} \in D$. As $\mathrm{desc}(D) \subseteq \mathrm{wdesc}(D)$ we have that $\mathbf{x} \in \mathrm{wdesc}(D)$. Since C is t-wTA, then $\mathbf{y} \in D$ and hence C is also t-TA. □

The converse of the claim is not true, as the next example demonstrates.

Example 5.2.1. The code $C = \{11002, 10111, 22212\}$ over the alphabet $Q = \{0, 1, 2\}$ is a 2-TA code, but not a 2-wTA, as the code is not trivial.

A summary of the relations amongst traceability and IPP codes that have been proved so far, is given in Figure 5.3.

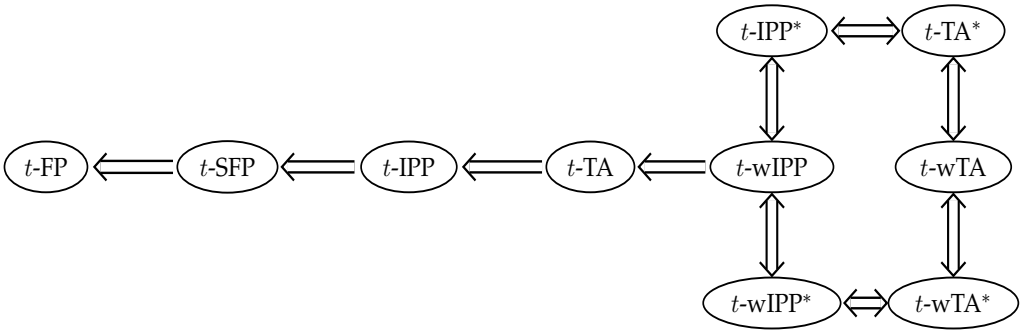


Figure 5.3: Relations of narrow-sense frameproof and secure frameproof codes with traceability and IPP codes under all four models of descendant set.

5.3 Frameproof and Secure Frameproof Codes

This section investigates the connections amongst frameproof and secure frameproof codes under the remaining models of descendant sets. Moreover, results are presented that connect these three models with the results of section 5.1 on the narrow-sense model. The cases of frameproof and secure frameproof are studied separately.

First, we examine the relation between t-wSFP and t-wFP codes. The proof is the same as the proof of Proposition 5.1.3 on t-SFP and t-FP by Stinson et al., apart from the different descendant set that is used, and is included here for completeness.

Proposition 5.3.1. *A t-wSFP code is a t-wFP code.*

Proof. Let C be a t-wSFP code and assume for a contradiction that it is not t-wFP. This means that there exists a set $D \subseteq C$ of size a most t, for which the intersection between the code and the set $\mathrm{wdesc}(D)$ is different from D.

Note that as $D \subseteq \mathrm{wdesc}(D)$, there must exist a word $\mathbf{y} \notin D$, but $\mathbf{y} \in C$ and $\mathbf{y} \in \mathrm{wdesc}(D)$. Let $\{\mathbf{y}\} = D'$, then $D \cap D' = \emptyset$, but $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$, which is a contradiction. □

The following example shows that the converse of the claim is not true.

Example 5.3.1. The code $C = \{01210, 10310, 00100, 00011\}$ is 2-wFP, over the alphabet $Q = \{0, 1, 2, 3\}$. Let $D = \{01210, 10310\}$ and $D' = \{00100, 00011\}$ be subsets of the code. Then, the word $\mathbf{x} = 00010$ belongs to the intersection of $\mathrm{wdesc}(D)$ and $\mathrm{wdesc}(D')$, while $D \cap D'$ is empty. Thus, C is not 2-wSFP.

5.3.1 Secure Frameproof Codes

Next, the relations between secure frameproof codes are presented, under different types of descendant sets.

Proposition 5.3.2. *A code C is t-SFP* if and only if C is a t-SFP code.*

Proof. Assume that C is t-SFP*, which means that for all distinct pairs of subsets D, D' with $|D| \leq t$, $|D'| \leq t$, if $D \cap D' = \emptyset$ then $\mathrm{desc}^*(D) \cap \mathrm{desc}^*(D') = \emptyset$. Note, that $\mathrm{desc}(D) \subseteq \mathrm{desc}^*(D)$ and $\mathrm{desc}(D') \subseteq \mathrm{desc}^*(D')$ imply that $\mathrm{desc}(D) \cap \mathrm{desc}(D') \subseteq \mathrm{desc}^*(D) \cap \mathrm{desc}^*(D')$ and since $\mathrm{desc}^*(D) \cap \mathrm{desc}^*(D') = \emptyset$, we also have that $\mathrm{desc}(D) \cap \mathrm{desc}(D') = \emptyset$ when $D \cap D' = \emptyset$. Hence, C is a t-SFP code.

For the reverse direction, assume that C is t-SFP. Using the fact that for every two subsets D, D' of C, $\mathrm{desc}(D) \cap \mathrm{desc}(D') \subseteq \mathrm{desc}^*(D) \cap \mathrm{desc}^*(D')$, we get $\mathrm{desc}^*(D) \cap \mathrm{desc}^*(D') \neq \emptyset$ whenever $\mathrm{desc}(D) \cap \mathrm{desc}(D') \neq \emptyset$. Since by the definition of t-SFP* code, $\mathrm{desc}(D) \cap \mathrm{desc}(D') \neq \emptyset$ imply $D \cap D' \neq \emptyset$, we automatically have that $\mathrm{desc}^*(D) \cap \mathrm{desc}^*(D') \neq \emptyset$ imply $D \cap D' \neq \emptyset$, which means that C is t-SFP*. □

Proposition 5.3.3. *A code C is t-wSFP* if and only if C is a t-wSFP code.*

The following proof can be disregarded, as it is identical to the previous one, with the only change being the type of descendant set. It is presented here for completeness.

Proof. Assume that C is t-wSFP*, which means that for all distinct pairs of subsets D, D' with $|D| \leq t, |D'| \leq t$, if $D \cap D' = \emptyset$ then $\mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D') = \emptyset$. Note, that the relations between the descendant sets, namely $\mathrm{wdesc}(D) \subseteq \mathrm{wdesc}^*(D)$ and $\mathrm{wdesc}(D') \subseteq \mathrm{wdesc}^*(D')$, imply that $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \subseteq \mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D')$ and since $\mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D') = \emptyset$, we also have that $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') = \emptyset$ when $D \cap D' = \emptyset$. Hence, C is a t-wSFP code.

For the opposite direction, assume that C is t-wSFP. Using the fact that for every two subsets D, D' of C, $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \subseteq \mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D')$, we get $\mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D') \neq \emptyset$ if $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$. Since by the definition of t-wSFP* code, $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$ imply $D \cap D' \neq \emptyset$, we automatically have that $\mathrm{wdesc}^*(D) \cap \mathrm{wdesc}^*(D') \neq \emptyset$ imply $D \cap D' \neq \emptyset$, which means that C is t-wSFP*. □

In contrast to the above propositions, the relation between t-SFP and t-wSFP codes is one way.

Proposition 5.3.4. *A t-wSFP code is a t-SFP code.*

Proof. Let C be a t-wSFP code. Then, observing that $\mathrm{desc}(D) \subseteq \mathrm{wdesc}(D)$ and $\mathrm{desc}(D') \subseteq \mathrm{wdesc}(D')$, leads to $\mathrm{desc}(D) \cap \mathrm{desc}(D') \subseteq \mathrm{wdesc}(D) \cap \mathrm{wdesc}(D')$ and the claim follows directly. Specifically, by the definition of t-wSFP code, $D \cap D' = \emptyset$ implies $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') = \emptyset$, which means that when $D \cap D' = \emptyset$, then also $\mathrm{desc}(D) \cap \mathrm{desc}(D') = \emptyset$. Hence, C is t-SFP. □

The next example shows that there are t-SFP codes which are not t-wSFP codes, and hence the validity of the converse of the previous claim does not hold.

Example 5.3.2. Let us examine the 2-SFP code $C = \{0000, 0111, 1120, 2021\}$. Assume that it is also a 2-wSFP code and take the subsets $D = \{0000, 0111\}$ and $D = \{1120, 2021\}$. According to the definition of t-wSFP code, the empty intersection of D and D' implies that $\mathrm{wdesc}(D)$ and $\mathrm{wdesc}(D')$ do not intersect, as well. The word $\mathbf{x} = 0120$ is a member of the wide-sense descendant set of both D and D', and thus $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$, which contradicts the assumption that C is 2-wSFP. Hence, the code is not 2-wSFP.

5.3.2 Frameproof Codes

The connections between the different frameproof codes are similar to those amongst the secure frameproof codes, as the following propositions prove.

Proposition 5.3.5. *A code C is t-FP* if and only if C is a t-FP code.*

Proof. Let C be a t-FP code, that is, for all subsets D of C with $|D| \leq t$, we have $\mathrm{desc}(D) \cap C = D$. The expanded narrow-sense descendant sets, apart from the words formed with letters of the codewords in D, also consist of words that contain the symbol '?'. As the code C does not include words with '?', the intersection between $\mathrm{desc}^*(D)$ and C has to contain words without unreadable or deleted marks. This implies that for all $D \subseteq C$, we have $\mathrm{desc}^*(D) \cap C = \mathrm{desc}(D) \cap C$, which indicates that C is both t-FP and t-FP* code. □

Proposition 5.3.6. *A code C is t-wFP* if and only if C is a t-wFP code.*

The proof is the same as the previous proof and is included here for completeness.

Proof. For all $D \subseteq C$, the set $\mathrm{wdesc}^*(D)$ contains words with unreadable or deleted marks, along with words composed by alphabet letters. Therefore, the intersection $\mathrm{wdesc}^*(D) \cap C$ consists of words without the symbol '?', which leads to $\mathrm{wdesc}^*(D) \cap C = \mathrm{wdesc}(D) \cap C$. Hence, C being t-wFP, is

equivalent to C being t-wFP*, as $\mathrm{wdesc}(D) \cap C = D = \mathrm{wdesc}^*(D) \cap C$, for every $D \subseteq C$ of size at most t. □

Similar to the connection between t-wSFP and t-SFP codes, is the connection of t-wFP and t-FP codes.

Proposition 5.3.7. *A t-wFP code is a t-FP code.*

Proof. Let C be a t-wFP code, then for every $D \subseteq C$ with $|D| \leq t$, we have $\mathrm{wdesc}(D) \cap C = D$. Notice, that $D \subseteq \mathrm{desc}(D) \subseteq \mathrm{wdesc}(D)$. This means that $\mathrm{wdesc}(D) \cap C = D$ implies $\mathrm{desc}(D) \cap C = D$, for all subsets D of C of size at most t, or in other words, C is t-FP. □

The following example shows that the opposite direction is not true.

Example 5.3.3. The code $C = \{10021, 00111, 20201\}$ is a 2-FP code over the alphabet $Q = \{0, 1, 2\}$, but not 2-wFP: for the subset $D = \{10021, 00111\}$, we have

$$\begin{aligned}\mathrm{wdesc}(D) = \{&00001, 00011, 00021, 00101, 00111, 01021, 00201, 00211, 00221,\\ &10001, 10011, 10021, 10101, 10111, 11021, 10201, 10211, 10221,\\ &20001, 20011, 20021, 20101, 20111, 21021, 20201, 20211, 20221\}\end{aligned}$$

and $\mathrm{wdesc}(D) \cap C = C \neq D$.

5.3.3 Unifying the Relations Between Fingerprinting Codes

In order to bridge the results on frameproof and secure frameproof codes with those in Figure 5.3, it remains to examine the way t-wSFP and t-wIPP codes are related.

Proposition 5.3.8. *A t-wIPP code is a t-wSFP code.*

The proof is similar to the proof of Proposition 5.1.2, except for the different descendant model and is included only for completeness.

Proof. Let C be a t-wIPP code and D, D' subsets of C with $|D| \leq t$ and $|D'| \leq t$. Additionally, let $\mathbf{x}$ be a word in the intersection of $\mathrm{wdesc}(D)$ and $\mathrm{wdesc}(D')$. Thus, we have D, $D' \in \mathcal{P}^w_{t,C}(\mathbf{x})$. Then, as C is t-wIPP, there exists $\mathbf{y} \in \bigcap_{D_i \in \mathcal{P}^w_{t,C}(\mathbf{x})} D_i$. Notice, that

$$\bigcap_{D_i \in \mathcal{P}^w_{t,C}(\mathbf{x})} D_i \subseteq D \cap D',$$

and consequently $\mathbf{y} \in D \cap D'$. Hence, for all distinct D, D' subsets of C of size at most t, when $\mathrm{wdesc}(D) \cap \mathrm{wdesc}(D') \neq \emptyset$ then $D \cap D' \neq \emptyset$, which proves that C is t-wSFP. □

The next example presents a code which is 2-wSFP but not 2-wIPP, proving that the converse of the claim above is not true.

Example 5.3.4. The code $C = \{1002, 1201, 2001, 2212\}$ is a non trivial 2-wSFP code and hence cannot be 2-wIPP.

The final picture of all the relations amongst the fingerprinting codes that have been proved in this chapter, is given by Figure 5.4. Note, that when the relation between two codes is denoted by the implication symbol $(\Rightarrow)$, is silently indicated that there exists an example showing that the opposite direction does not hold.

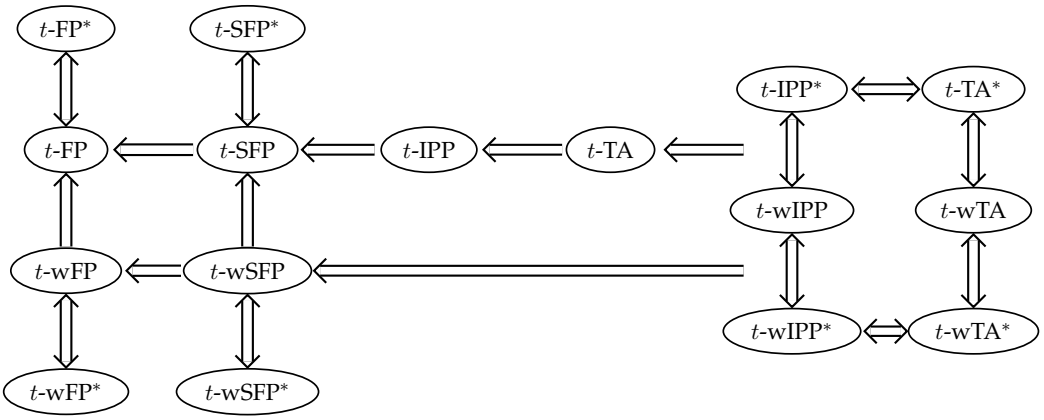


Figure 5.4: Relations of narrow-sense frameproof and secure frameproof codes and traceability and IPP codes under all four models of descendant set.

Chapter 6

Wide-Sense 2-Frameproof Codes

This chapter studies wide-sense frameproof codes and aims to establish a good upper bound for the maximal size of such codes. The first section consists of preliminaries on these codes, most of which relate to the minimum distance of a code. Using these preliminaries, the section two presents results on the maximal size of wide-sense frameproof codes of small length. The main result of this part of the thesis (Theorem 6.3.8) is presented in the third section. The section studies the case of arbitrary length and provides an upper bound on the maximal size of wide-sense frameproof codes, which improves upon the bound in [50] by Stinson and Wei. Our bound, like that of Stinson and Wei, does not depend on the alphabet size. Additionally, we include a separate section that uses the techniques from the arbitrary length case to obtain the maximal wide-sense frameproof code of length 5, on any alphabets of size q. In particular, we prove that in the case where $q = 2$ or 3 the largest such code contains 6 codewords, whereas for $q \geq 4$ the maximal size is 8.

6.1 Properties of 2-wFP codes

We begin this section with a proposition which can be easily deduced from Definition 3.3.3 of the wide-sense frameproof codes, and then we continue with some properties regarding the minimum distance of the codes and the common positions of the codewords.

Proposition 6.1.1 (**Property of 2-wFP codes**). *A code C of length ℓ is 2-wFP if and only if for all distinct words $\mathbf{u}, \mathbf{v}, \mathbf{z} \in C$ there exists a coordinate $i \in \{1, \ldots, \ell\}$*

such that $u_i = v_i \neq z_i$.

Lemma 6.1.2. *The minimum distance of a 2-wFP code of length* $\ell \geq 2$ *satisfies the following inequality:*

$$1 \leq d(C) \leq \ell - 1.$$

Proof. When $\ell = 2$, it follows from Proposition 6.1.1 that we must have $d(C) = 1$. In the case where $\ell > 2$, Proposition 6.1.1 implies that no two codewords can disagree in all positions, because the corresponding wide-sense descendant set would consist of all words from the alphabet Q. Thus, the minimum distance, $d(C)$, of a 2-wFP code C of length ℓ containing at least three words, is at most $\ell - 1$. Furthermore, $d(C)$ cannot be zero, as we are considering distinct codewords. □

Proposition 6.1.3. *Let* $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\}$ *be a 2-wFP code of length* ℓ *and size* $|C| = m$. *If there exists a position* $i \in \{1, \ldots, \ell\}$ *where* $c_i^1 = c_i^2 = \ldots = c_i^m$, *then there exists a 2-wFP code* C' *of length* $\ell - 1$ *and size* $|C'| = m$.

Proof. Without loss of generality, let the common position i be the first:

$$c_1^1 = c_1^2 = \ldots = c_1^m.$$

By hypothesis, the code C is 2-wFP, thus for all distinct $j, j', j'' \in \{1, \ldots, m\}$ there exists a position i such that $c_i^j = c_i^{j'} \neq c_i^{j''}$. Since i is not 1, as it would violate the 2-wFP property of the code C, we can remove the first coordinate from all the codewords and still retain the 2-wFP property. Hence, we are left with a 2-wFP code C' of length $\ell - 1$ and size m. □

Next, we study the behaviour of wide-sense 2-frameproof codes with regard to the size of the codes. In particular, we examine the changes of the size, when the minimum distance takes its extreme values.

Proposition 6.1.4. *If the minimum distance* $d(C)$ *of a 2-wFP code* C *of length* ℓ *is 1, then* $|C| \leq 2$.

Proof. Suppose that $|C| = 3$ with $C = \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$. Since $d(C) = 1$, there exists a pair of codewords, say $\mathbf{u}$ and $\mathbf{v}$, such that $d(\mathbf{u}, \mathbf{v}) = 1$. Without loss of generality, assume that $\mathbf{u}$ and $\mathbf{v}$ agree in the first $\ell - 1$ positions. From the property of 2-wFP codes (Proposition 6.1.1) there exist distinct positions $i, j \in \{1, \ldots, \ell\}$, such that $w_i = u_i \neq v_i$ and $w_j = v_j \neq u_j$. The choices for i and j are the positions where $\mathbf{u}$ and $\mathbf{v}$ disagree, but since $d(\mathbf{u}, \mathbf{v}) = 1$, i and j must be both equal to l, which is a contradiction. Since every 2-wFP code of size $|C| > 2$ contains a 3-subset, Remark 3.3.27 concludes the proof. □

Proposition 6.1.5. *If the minimum distance $d(C)$ of a 2-wFP code C of length $\ell > 2$ is $d(C) = \ell - 1$, then $|C| \leq \ell + 1$.*

Proof. Since C is 2-wFP and by Lemma 6.1.2 there do not exist two codewords of distance ℓ, $d(C) = \ell - 1$ implies that C is equidistant. This means that each pair of codewords agree in exactly one position. We begin with a 2-wFP code containing only two words and gradually add more words until one of the conditions, either the 2-wFP property (Proposition 6.1.1) or the distance restriction (Lemma 6.1.2), cease to hold. Let $C = \{\mathbf{c}, \mathbf{c^1}\}$ and without loss of generality, assume $c_1 = c_1^1$, since we know that $d(\mathbf{c}, \mathbf{c^1}) = \ell - 1$. We add a word $\mathbf{c^2} \in Q^\ell$. We claim that there cannot exist a position where all three words agree. Assume for a contradiction that $c_1 = c_1^1 = c_1^2$. In order for C to be 2-wFP, there must exist a position $i \neq 1$, such that $c_i = c_i^1 \neq c_i^2$, which is a contradiction to the code C being equidistant. Hence, $\mathbf{c}$, $\mathbf{c^1}$ and $\mathbf{c^2}$ cannot agree in the first position. Hence, by the 2-wFP property

$$\exists i \in \{2, \ldots, \ell\} \quad \text{such that} \quad c_i^2 = c_i \neq c_i^1.$$

Without loss of generality, let $i = 2$. Add a word $\mathbf{c^3} \in Q^\ell$. Applying the same constraints as above, the only choices for the position $j \in \{1, \ldots, \ell\}$ such that

$$c_j^3 = c_j \neq c_j^t \quad \text{for} \quad t = 1, 2,$$

are $j \in \{3, \ldots, \ell\}$. Without loss of generality take $j = 3$. Similarly, we have $c_4^4 = c_4$, $c_5^5 = c_5$ and so on, until we have used the ℓ^{th} position of $\mathbf{c}$, by adding the $\mathbf{c}^\ell$ word, for which $c_\ell^\ell = c_\ell$ and $c_\ell^\ell \neq c_\ell^t$ for all $t \in \{1, \ldots, \ell - 1\}$. Suppose we add one more word, $\mathbf{c}^{\ell+1}$. Then, as in the previous cases,

$$\exists j \in \{1, \ldots, \ell\} \quad \text{such that} \quad c_j^{\ell+1} = c_j \neq c_j^t \quad \forall t \in \{1, \ldots \ell\}$$

Since all possible values of j have already been used, assigning a value from $\{1, \ldots, \ell\}$ to j, leads to three words agreeing in the same position:

$$c_j^{\ell+1} = c_j = c_j^j.$$

This implies that codewords $\mathbf{c^j}$ and $\mathbf{c}$ agree in two positions, which is a contradiction. Hence, the word $\mathbf{c}^\ell$ is the last we can add, preserving at the same time the 2-wFP property and the distance restriction. Finally, counting the added words we obtain the maximum size of the 2-wFP code C:

$$|C| \leq 1 + \sum_{m=1}^{l} 1 = \ell + 1.$$

□

6.2 Small Length Case

This section examines each wide-sense 2-frameproof code separately, according to their length. The values of the lengths to be considered are $\ell = 2, 3, 4$. The of $\ell = 5$ is also included in the thesis, however it is discussed in a subsequent section, as the examination of this particular length requires further information on 2-wFP codes, which is obtained from the case of arbitrary length.

Example 6.2.1. Codes C_1, C_2 and C_3 are examples of 2-wFP codes of length $\ell = 2$, 3, 4, respectively.

$C_1 = \{10, 11\}$,

$C_2 = \{100, 010, 001, 111\}$,

$C_3 = \{1000, 0100, 0010, 0001, 1111\}$.

As the next propositions prove, the sizes of C_1, C_2 and C_3 are the best possible for the corresponding lengths. Furthermore, these codes show that the binary alphabet is sufficient for the codes to attain the maximum number of codewords.

Proposition 6.2.1. *Let C be a 2-wFP code of length $\ell = 2$. Then $|C| \leq 2$.*

Proof. By the distance restrictions of Lemma 6.1.2 we have that $d(C) = 1$ and by Proposition 6.1.4 the size of C cannot be more than 2. □

Proposition 6.2.2. *Let C be a 2-wFP code of length $\ell = 3$. Then $|C| \leq 4$.*

Proof. Assume $|C| > 2$. By Lemma 6.1.2 we have $1 \leq d(C) \leq 2$. Thus, we need to examine two different cases, $d(C) = 1$ or $d(C) = 2$. In the first case, by Proposition 6.1.4 we get $|C| \leq 2$ and in the second $|C| \leq \ell + 1 = 4$ (Proposition 6.1.5). Hence, the size of C is at most $\ell + 1 = 4$. □

Proposition 6.2.3. *Let C be a 2-wFP code of length $\ell = 4$. Then $|C| \leq 5$.*

Proof. Assume $|C| > 2$. By Lemma 6.1.2 we restrict to the cases where $1 \leq d(C) \leq 3$ and by Propositions 6.1.4 and 6.1.5 we obtain the following:

- If $d(C) = 1$ then $|C| \leq 2$.
- If $d(C) = 3$ then $|C| \leq 5$.

Thus, the only case to examine is when $d(C) = 2$. Assume for a contradiction, that $C = \{\mathbf{c^1}, \mathbf{c^2}, \mathbf{c^3}, \mathbf{c^4}, \mathbf{c^5}, \mathbf{c^6}\}$ is of size $\ell + 2 = 6$. Since $d(C) = 2$, there exists a pair of codewords, say $\mathbf{c^1}$ and $\mathbf{c^2}$, such that $d(\mathbf{c^1}, \mathbf{c^2}) = 2$. Without loss of generality assume that $\mathbf{c^1}$ and $\mathbf{c^2}$ agree in the first two positions: $c_1^1 = c_1^2$ and $c_2^1 = c_2^2$. By the property of 2-wFP codes (Proposition 6.1.1) we know that for every triple $\{\mathbf{c^1}, \mathbf{c^2}, \mathbf{c^j}\}$, where $\mathbf{c^j} \in C \setminus \{\mathbf{c^1}, \mathbf{c^2}\}$, there exist distinct positions $i, k \in \{1, \ldots, \ell\}$ such that

$$c_i^j = c_i^1 \neq c_i^2 \quad \text{and} \quad c_k^j = c_k^2 \neq c_k^1.$$

Since $\mathbf{c^1}$ and $\mathbf{c^2}$ agree on their first two components, the only choices for i and k are the third and fourth position: $(i,k) \in \{(3,4),(4,3)\}$. Hence, taking all the 3-subsets of C including $\mathbf{c^1}$ and $\mathbf{c^2}$, the choices for the coordinates that we can use in order to satisfy the 2-wFP property, are the third and fourth. The number of 3-subsets containing $\mathbf{c^1}$ and $\mathbf{c^2}$ is 4, namely $\{\mathbf{c^1},\mathbf{c^2},\mathbf{c^3}\}$, $\{\mathbf{c^1},\mathbf{c^2},\mathbf{c^4}\}$, $\{\mathbf{c^1},\mathbf{c^2},\mathbf{c^5}\}$ and $\{\mathbf{c^1},\mathbf{c^2},\mathbf{c^6}\}$. Without loss of generality, by choosing $\mathbf{c^1} = 1111$ and $\mathbf{c^2} = 1100$, we obtain the following cases:

Case 1: $(i,k) = (3,4)$, for three of the words in the set $\{\mathbf{c^3},\mathbf{c^4},\mathbf{c^5},\mathbf{c^6}\}$, say $\mathbf{c^3},\mathbf{c^4}$ and $\mathbf{c^5}$:

$$\begin{array}{c||c} \mathbf{c^1} & 1\,1\,1\,1 \\ \mathbf{c^2} & 1\,1\,0\,0 \\ \mathbf{c^3} & \_\,\_\,1\,0 \\ \mathbf{c^4} & \_\,\_\,1\,0 \\ \mathbf{c^5} & \_\,\_\,1\,0 \\ \mathbf{c^6} & \_\,\_\,\_\,\_ \end{array}$$

Case 2: $(i,k) = \begin{cases} (4,3), & \text{for two of the words of } \{\mathbf{c^3},\mathbf{c^4},\mathbf{c^5},\mathbf{c^6}\} \\ (3,4), & \text{for the other two.} \end{cases}$

Without loss of generality assume the following:

$$\begin{array}{c||c} \mathbf{c^1} & 1\,1\,1\,1 \\ \mathbf{c^2} & 1\,1\,0\,0 \\ \mathbf{c^3} & \_\,\_\,0\,1 \\ \mathbf{c^4} & \_\,\_\,0\,1 \\ \mathbf{c^5} & \_\,\_\,1\,0 \\ \mathbf{c^6} & \_\,\_\,1\,0 \end{array}$$

In the case 1, since C is 2-wFP we know that the subset $\{\mathbf{c^4},\mathbf{c^5},\mathbf{c^6}\}$ also forms a 2-wFP code. By the 2-wFP property, there exist distinct $i,j,k \in \{1,\ldots,\ell\}$ such that

$$c_i^4 = c_i^5 \neq c_i^6, \quad c_j^4 = c_j^6 \neq c_j^5 \quad \text{and} \quad c_k^5 = c_k^6 \neq c_k^4$$

Since all three words agree in the last two positions, there are only two choices for i, j, k, namely the first and the second position. This means that using only the first two positions of $\mathbf{c^4}, \mathbf{c^5}$ and $\mathbf{c^6}$, we have formed a 2-wFP code. But this contradicts Proposition 6.2.1, which states that the maximum size of a 2-wFP code of length $\ell = 2$ is 2. Hence, in the case 1, the best possible size of a 2-wFP code is $5 = \ell + 1$.

For the case 2, let us consider the triples $\{\mathbf{c^3}, \mathbf{c^4}, \mathbf{c^j}\}$ for $j = 1, 2, 5, 6$. According to the 2-wFP property, for every such triple there exist distinct positions (s, r) such that

$$c_s^j = c_s^3 \neq c_s^4 \quad \text{and} \quad c_r^j = c_r^4 \neq c_r^3$$

Clearly, $(s, r) \in \{(1, 2), (2, 1)\}$. By assumption, codewords $\mathbf{c^1}$ and $\mathbf{c^2}$ are determined, thus we either have $c_1^3 c_2^4 = c_1^1 c_2^2 = 11$ or $c_2^3 c_1^4 = c_2^1 c_1^2 = 11$. Due to symmetry, assume $c_1^3 c_2^4 = 11$:

$$\begin{array}{c||l} \mathbf{c^1} & 1\,1\,1\,1 \\ \mathbf{c^2} & 1\,1\,0\,0 \\ \mathbf{c^3} & 1\,\_\,0\,1 \\ \mathbf{c^4} & \_\,1\,0\,1 \\ \mathbf{c^5} & \_\,\_\,1\,0 \\ \mathbf{c^6} & \_\,\_\,1\,0 \end{array}$$

Similarly, we next examine the relation between $\mathbf{c^3}$, $\mathbf{c^4}$, with codewords $\mathbf{c^5}$ and $\mathbf{c^6}$. If for $\mathbf{c^5}$ we have $(s, r) = (1, 1)$, which means that $c_1^5 = c_1^3 = 1$ and $c_2^5 = c_1^4 = 1$, then there exist three codewords, namely $\mathbf{c^1}$, $\mathbf{c^2}$, $\mathbf{c^5}$, that agree in the first two positions. As a subset of a 2-wFP code, $\{\mathbf{c^1}, \mathbf{c^2}, \mathbf{c^5}\}$ is also 2-wFP. However, by deleting the common positions of these three words, the remaining code has length 2 and size 3 and by Proposition 6.2.1 it cannot be a 2-wFP code, a contradiction. Hence $c_1^5 \neq 1$ and $c_2^5 \neq 1$. Similarly, the same restriction holds for $\mathbf{c^6}$, as well. Thus, the only choice for $\mathbf{c^5}$ and $\mathbf{c^6}$ to agree with $\mathbf{c^3}$ and $\mathbf{c^4}$, and follow at the same time the property of the 2-wFP code,

is the following:

$$c_1^5 = c_1^4 \quad \text{and} \quad c_1^6 = c_1^4$$
$$c_2^5 = c_2^3 \quad \text{and} \quad c_2^6 = c_2^3$$

This means that the two codewords are identical:

$$\begin{aligned} \mathbf{c^5} &= c_1^4\, c_2^3\, 0\, 1 \\ \mathbf{c^6} &= c_1^4\, c_2^3\, 0\, 1, \end{aligned}$$

which is a contradiction, as all codewords are distinct.

Since both cases led to contradiction, we conclude that there do not exist 2-wFP codes of length $l = 4$ and size $|C| > 5$. □

6.3 Arbitrary Length Case

The aim of this section is to study the case of the wide-sense 2-frameproof codes with arbitrary, but fixed length ℓ and as a result, to present an improved upper bound on their size, compared to the bound of Theorem 5.2 [50] by Stinson and Wei. As mentioned in [50], the structure of these codes is closely related to Sperner families. Here, we formally define this relation and moreover, present additional properties, which lead to a major improvement on the size of the code, in particular cases.

We first begin by defining the *coincidence function*, which on input of two words gives the set of positions where these words coincide. As a following theorem shows (Theorem 6.3.1), one way of obtaining such a function is by considering a code and defining this function to return the common positions of pairs of codewords.

Let $M = \{1, \ldots, m\}$ and $I : M \times M \to \mathcal{P}(\{1, \ldots, \ell\})$ satisfy the following properties:

1. For all $j_1, j_2 \in M \quad I(j_1, j_2) = I(j_2, j_1)$

2. (a) For all $j \in M \quad I(j, j) = \{1, \ldots, \ell\}$

(b) For all distinct $j_1, j_2 \in M \quad I(j_1, j_2) \neq \{1, \ldots, \ell\}$

3. For all $j_1, j_2, j_3 \in M$ and $i \in \{1, \ldots, \ell\}$, if $i \in I(j_1, j_2)$ and $i \in I(j_2, j_3)$ then $i \in I(j_1, j_3)$

Using the above properties of the coincidence function I, we define an equivalence relation on the set M. For every $i \in \{1, \ldots, \ell\}$, we say that j_1 *is equivalent to* j_2 *with respect to* i *if and only if* $i \in I(j_1, j_2)$. We denote this relation by $j_1 \underset{i}{\sim} j_2$. It is easy to show that the defined relation is an equivalence relation:

(a) Let $j_1, j_2 \in M$, $i \in \{1, \ldots, \ell\}$ and $j_1 \underset{i}{\sim} j_2$. By definition, $i \in I(j_1, j_2)$ and using property 1 of the coincidence function, we have that $i \in I(j_2, j_1)$, which is equivalent by definition to $j_2 \underset{i}{\sim} j_1$. Thus, for all $j_1, j_2 \in M$ $j_1 \underset{i}{\sim} j_2$ is equivalent to $j_2 \underset{i}{\sim} j_1$ and thus the $\underset{i}{\sim}$-relation is symmetric.

(b) Let $j \in M$, $i \in \{1, \ldots, \ell\}$ and $j \underset{i}{\sim} j$. By definition, $i \in I(j, j) = \{1, \ldots, \ell\}$ for all $j \in M$ (property 2(a)). In other words, the $\underset{i}{\sim}$-relation is reflexive.

(c) Let $j_1, j_2, j_3 \in M$, $i \in \{1, \ldots, \ell\}$ and $j_1 \underset{i}{\sim} j_2$ and $j_2 \underset{i}{\sim} j_3$. Then by definition, i belongs in both $I(j_1, j_2)$ and $I(j_2, j_3)$, which from the third property of the coincidence function is equivalent to $i \in I(j_1, j_3)$. Thus we have proven that the $\underset{i}{\sim}$-relation is transitive: for all $j_1, j_2, j_3 \in M$ if $j_1 \underset{i}{\sim} j_2$ and $j_2 \underset{i}{\sim} j_3$, then $j_1 \underset{i}{\sim} j_3$.

Theorem 6.3.1. *A function I is a coincidence function if and only if there exists an alphabet $Q = \{0, 1, \ldots, q-1\}$ of size $q \in \mathbb{N}$ and a code $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\} \subseteq Q^\ell$, such that for every $j_1, j_2 \in \{1, \ldots, m\}$, $I(j_1, j_2) = \{i \in \{1, \ldots, \ell\} : c_i^{j_1} = c_i^{j_2}\}$.*

Proof. We begin by proving the reverse direction of the claim. Let $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\} \subseteq Q^\ell$ be a code and for all $j_1, j_2 \in \{1, \ldots, m\}$ let $I(j_1, j_2)$ be the set of positions where the codewords $\mathbf{c^{j_1}}$ and $\mathbf{c^{j_2}}$ agree. We prove that $I(j_1, j_2)$ is the coincidence function, by showing that for all $j_1, j_2 \in \{1, \ldots, m\}$, $I(j_1, j_2)$ has the properties 1, 2 and 3.

We can easily see that $i \in I(j_1, j_2)$ if and only if $i \in I(j_2, j_1)$, since both relations indicate that the codewords $\mathbf{c^{j_1}}$ and $\mathbf{c^{j_2}}$ agree in position i. Hence $I(j_1, j_2) = I(j_2, j_1)$. When $j_1 = j_2$, then $\mathbf{c^{j_1}}$ agrees with $\mathbf{c^{j_2}}$ in every position, thus $I(j_1, j_2) = \{1, \ldots, \ell\}$. When $j_1 \neq j_2$, then there exists a position i where $\mathbf{c^{j_1}}$ and $\mathbf{c^{j_2}}$ disagree, which implies that $I(j_1, j_2) \neq \{1, \ldots, \ell\}$. Hence, $j_1 = j_2$ if and only if $I(j_1, j_2) = \{1, \ldots, \ell\}$, which summarises the properties 2(a) and 2(b) of the coincidence function. To prove that I also satisfies the third property, let $i \in I(j_1, j_2)$ and $i \in I(j_2, j_3)$. Then by the definition of the sets $I(j_1, j_2)$ and $I(j_2, j_3)$, we have that $c_i^{j_1} = c_i^{j_2}$ and $c_i^{j_2} = c_i^{j_3}$ respectively, which lead to $c_i^{j_1} = c_i^{j_3}$. The latter equality is equivalent to $i \in I(j_1, j_3)$ and the first part of the proof is completed.

For the other direction, given the coincidence function I our aim is to prove the existence of a set $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\} \subseteq Q^\ell$, such that for every $j_1, j_2 \in \{1, \ldots, m\}$ we have $I(j_1, j_2) = \{i \in \{1, \ldots, \ell\} : c_i^{j_1} = c_i^{j_2}\}$. First, we apply the $\underset{i}{\sim}$-equivalence relation to the set $\{1, \ldots, m\}$ and partition it into equivalence classes:

$$\{1, \ldots, m\} = X_0^{(i)} \cup X_1^{(i)} \cup \ldots \cup X_{r_i}^{(i)}.$$

Since the partition depends on i, for every $i \in \{1, \ldots, \ell\}$ we have a collection of equivalence classes $X_r^{(i)}$, $r = 0, \ldots, r_i$. We define the size of the alphabet Q to be $q = \max\limits_{1 \leq i \leq \ell} r_i$ and to each equivalence class we assign a letter from Q. As for $i \neq i'$ the sets $X_r^{(i)}$ and $X_r^{(i')}$ are independent, we can assign the same letters to equivalence classes that correspond to different values of i. For $j \in \{1, \ldots, m\}$ define $\mathbf{c^j} = c_1^j \ldots c_\ell^j \in Q^\ell$, by $c_i^j = \alpha$ if and only if $j \in X_\alpha^{(i)}$.

To complete the proof, we need to show that the words of the code satisfy the following property:

$$\forall j_1, j_2 \in \{1, \ldots, m\} \quad I(j_1, j_2) = \{i \in \{1, \ldots, \ell\} : c_i^{j_1} = c_i^{j_2}\}$$

and that the indices j_1, j_2 are distinct if and only if the corresponding codewords $\mathbf{c^{j_1}}$, $\mathbf{c^{j_2}}$ are also distinct.

By construction of the code C, it follows that for all $j \in X_\alpha^{(i)}$ $c_i^j = \alpha$. In other words, all codewords whose index j belongs to the same equivalence class $X_\alpha^{(i)}$, agree in position i. Now, let $i \in I(j_1, j_2)$. Then, by definition of the equivalence relation we have $i \in I(j_1, j_2)$ if and only if $j_1 \underset{i}{\sim} j_2$, which means that j_1, j_2 belong to the same equivalence class, say $X_\alpha^{(i)}$. Thus, the corresponding codewords agree in position i: $c_i^{j_1} = c_i^{j_2} = \alpha$. Furthermore, if $i \notin I(j_1, j_2)$ then $j_1 \underset{i}{\nsim} j_2$, which means that j_1 and j_2 belong to different equivalence classes, say $j_1 \in X_\alpha^{(i)}$ and $j_2 \in X_\beta^{(i)}$. This implies that $c_i^{j_1} = \alpha$ and $c_i^{j_2} = \beta$, in other words $c_i^{j_1} \neq c_i^{j_2}$. This proves that $I(j_1, j_2) = \{i \in \{1, \ldots, \ell\} : c_i^{j_1} = c_i^{j_2}\}$.

Finally, it remains to show that distinct indices j, $j' \in \{1, \ldots, m\}$ correspond to distinct codewords $\mathbf{c^j}$, $\mathbf{c^{j'}}$. Let $j \neq j'$ and assume for a contradiction, that there exists a pair $\mathbf{c^j}$, $\mathbf{c^{j'}} \in C$ such that for all $i \in \{1, \ldots, \ell\}$ $c_i^j = c_i^{j'}$. But this is equivalent to $j \underset{i}{\sim} j'$ for all $i \in \{1, \ldots, \ell\}$, which by definition is true, if and only if for all $i \in \{1, \ldots, \ell\}$ we have $i \in I(j, j')$. In other words, $I(j, j') = \{1, \ldots, \ell\}$. Now using property (2) of the coincidence function we get $j = j'$, a contradiction. Hence, $j \neq j'$ if and only if $c^j \neq c^{j'}$ and the proof of the theorem is concluded. □

Having defined the coincidence function, the next step is to show the way this function is connected to 2-wFP codes. Let C be a 2-wFP code of length ℓ and size m. For $\mathbf{c^j} \in C$ define the following family

$$\mathfrak{X}_j = \{I(j, j') : j' \in \{1, \ldots, m\} \setminus \{j\}\}. \tag{6.3.1}$$

Lemma 6.3.2. *Let $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\}$ be a code of length ℓ and size m, over an the alphabet Q of size q. Then, C is 2-wFP if and only if for all $j \in \{1, \ldots, m\}$ the families $\mathfrak{X}_j$ are Sperner.*

Proof. Let $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\}$ be a 2-wFP code. Then, for all $\{\mathbf{c^j}, \mathbf{c^{j'}}\} \in C$ we have that $\mathrm{wdesc}(\{\mathbf{c^j}, \mathbf{c^{j'}}\}) \cap C = \{\mathbf{c^j}, \mathbf{c^{j'}}\}$. Suppose for a contradiction, that there exists $j \in \{1, \ldots, m\}$ such that the corresponding family $\mathfrak{X}_j$ is not

Sperner. In other words, there exist distinct j_1, $j_2 \in \{1, \ldots, m\} \setminus \{j\}$ such that $I(j, j_1) \subseteq I(j, j_2)$. This means, that every $i \in I(j, j_1)$ is also a member of $I(j, j_2)$ and by definition of the coincidence function, for all i such that $c_i^j = c_i^{j_1}$, we also have $c_i^j = c_i^{j_2}$. Now, the wide-sense descendant set of $\mathbf{c^j}$, $\mathbf{c^{j_1}}$ consists of all words $\mathbf{x} = x_1 x_2 \ldots x_\ell \in Q^\ell$ such that

$$x_i = \begin{cases} c_i^j, & \text{if } c_i^j = c_i^{j_1} \\ \alpha \in Q, & \text{otherwise,} \end{cases}$$

which is equivalent to

$$x_i = \begin{cases} c_i^j, & \text{if } i \in I(j, j_1) \\ \alpha \in Q, & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathbf{c^{j_2}} \in \mathrm{wdesc}(\{\mathbf{c^j}, \mathbf{c^{j_1}}\})$: for all $i \in I(j, j_1)$ we have $c_i^j = c_i^{j_1} = c_i^{j_2}$ and the remaining positions of $\mathbf{c^{j_2}}$ are alphabet letters. But this violates the 2-wFP property of the code C, because $\mathrm{wdesc}(\{\mathbf{c^j}, \mathbf{c^{j_1}}\}) \cap C = \{\mathbf{c^j}, \mathbf{c^{j_1}}, \mathbf{c^{j_2}}\}$. Hence, for all distinct j, j_1, $j_2 \in \{1, \ldots, m\}$ we have $I(j, j_1) \nsubseteq I(j, j_2)$, which means that $\mathfrak{X}_j$ is a Sperner family.

To prove the other direction of the claim, suppose that for all $j \in \{1, \ldots, m\}$ the family $\mathfrak{X}_j$ is Sperner, which means that for all distinct j_1, $j_2 \in \{1, \ldots, m\}$ $I(j, j_1) \nsubseteq I(j, j_2)$. Thus, there exists $i \in I(j, j_1)$, say i_0, such that $i_0 \notin I(j, j_2)$. All words $\mathbf{x} = x_1 x_2 \ldots x_\ell \in Q^\ell$ for which

$$x_i = \begin{cases} c_i^j, & \text{if } i \in I(j, j_1) \\ \alpha \in Q, & \text{otherwise} \end{cases}$$

belong to $\mathrm{wdesc}(\mathbf{c^j}, \mathbf{c^{j_1}})$. Since $c_{i_0}^{j_2} \neq c_{i_0}^{j_1} = c_{i_0}^j$, by definition of the wide-sense descendant, $\mathbf{c^{j_2}} \notin \mathrm{wdesc}(\mathbf{c^j}, \mathbf{c^{j_1}})$. Hence, for all 2-subsets $\{\mathbf{c^j}, \mathbf{c^{j_1}}\}$ of C, $\mathrm{wdesc}(\{\mathbf{c^j}, \mathbf{c^{j_1}}\}) \cap C = \{\mathbf{c^j}, \mathbf{c^{j_1}}\}$. □

The above lemma shows that the relation between $\mathfrak{X}_j$ and the code C, results in $\mathfrak{X}_j$ having properties that are derived from the the fact that C is a 2-wFP code. Hence, it is possible that the incomparability of the members of $\mathfrak{X}_j$ might not be the only property that the family possesses. The next step

is to further examine the relation with the frameproof codes and present additional properties of $\mathfrak{X}_j$, that will lead to an improvement of the bound on its size and consequently, on the size of C. In particular, it is easy to see that

$$|C| = |\mathfrak{X}_j| + 1. \tag{6.3.2}$$

Proposition 6.3.3. *Let C be a 2-wFP code of length ℓ, size m and $\mathbf{c^j} \in C$. Then $\mathfrak{X}_j$ is non 2-covering Sperner family.*

Proof. Assume for a contradiction that the Sperner family $\mathfrak{X}_j$ formed by $\mathbf{c^j} \in C$ is 2-covering. This means that there exist $\mathbf{c^{j_1}}$, $\mathbf{c^{j_2}} \in C$ such that $I(j, j_1) \cup I(j, j_2) = \{1, \ldots, \ell\}$. According to the 2-wFP property (Proposition 6.1.1), there exists $i \in \{1, \ldots, \ell\}$ such that $c_i^{j_1} = c_i^{j_2} \neq c_i^j$. This implies that $i \notin I(j, j_1)$ and $i \notin I(j, j_2)$, which contradict the fact that $\mathfrak{X}_j$ is 2-covering, as there would exist a position i that does not belong to the union of $I(j, j_1)$ and $I(j, j_2)$. Hence, for a 2-wFP code C and a fixed word $\mathbf{c^j} \in C$, the corresponding Sperner family $\mathfrak{X}_j$ must be non 2-covering. □

As previously proved, every 2-wFP code C is associated with a non 2-covering Sperner family, by fixing a codeword and listing all the sets indicating the common positions between every codeword with the fixed word. Now, let us examine the behaviour of the size of the code, in the case where one of the sets of the non 2-covering Sperner family consists of only one element. First, we prove a result on Sperner families containing a singleton, that can be easily derived from the definition of a Sperner family and the upper bound on its maximal size.

Proposition 6.3.4. *Let $\mathcal{F}$ be a Sperner family over the set $\{1, \ldots, n\}$. If there exists a set $F \in \mathcal{F}$ such that $|F| = 1$, then*

$$|\mathcal{F}| \leq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor} + 1.$$

Proof. According to the definition of a Sperner family (Definition 2.1.1), the set F cannot be contained in any other set of the family. Thus, it is equivalent

to say that $\mathcal{F}$ consists of F and a family $\mathcal{D}$ of sets over $\{1,\dots,n\} \setminus F$. As $\mathcal{D}$ must also be Sperner, Theorem 2.1.2 yields

$$|\mathcal{D}| \leq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor},$$

which leads to the desired bound

$$|\mathcal{F}| = |\mathcal{D}| + 1 \leq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor} + 1.$$

□

Combining this proposition with equation (6.3.2) the following corollary is easily inferred.

Corollary 6.3.5. *Let $C = \{\mathbf{c^1},\dots,\mathbf{c^m}\}$ be a 2-wFP code of length ℓ over the alphabet Q. If there exist codewords $\mathbf{c^j}$, $\mathbf{c^k}$ such that $|I(j,k)| = 1$, then*

$$m \leq \binom{\ell-1}{\lfloor \frac{\ell-1}{2} \rfloor} + 2. \tag{6.3.3}$$

An important observation that is implied by the previous corollary, is the fact that the size of the code does not depend on the choice of the fixed word. This provides a flexibility that leads to direct conclusions, as the following lemma demonstrates.

Lemma 6.3.6. *Let ℓ be an odd number and C a 2-wFP code of length ℓ and size m. Let $\mathfrak{X}_j$ be the non 2-covering Sperner family, generated by the codeword $\mathbf{c_j}$, which consists of sets of size $\frac{\ell-1}{2}$. If $\mathfrak{X}_j$ is not intersecting, then*

(a) there exists a codeword $\mathbf{c^{j'}}$ with $j' \neq j$, whose corresponding family $\mathfrak{X}_{j'}$ contains a singleton,

(b)

$$m \leq \binom{\ell-1}{\lfloor \frac{\ell-1}{2} \rfloor} + 2.$$

Proof. Let X_1, X_2 be subsets of the family $\mathfrak{X}_j$. By hypothesis, $|X_1| = |X_2| = \frac{\ell-1}{2}$ and let $X_1 \cap X_2 = \emptyset$, which imply that their union has size $\ell - 1$. Recall that $X_1 = I(j, j_1)$ and $X_2 = I(j, j_2)$, represent the sets of positions where the codeword $\mathbf{c^j}$ agrees with $\mathbf{c^{j_1}}$ and $\mathbf{c^{j_2}}$ respectively, for $\mathbf{c^{j_1}}, \mathbf{c^{j_2}} \in C \setminus \{\mathbf{c^j}\}$. Since $I(j, j_1) \cap I(j, j_2) = \emptyset$ and $|I(j, j_1) \cup I(j, j_2)| = \ell - 1$, there is only one position left for $\mathbf{c^{j_1}}$ to agree with $\mathbf{c^{j_2}}$ and so $|I(j_1, j_2)| = 1$. This means that the families $\mathfrak{X}_{j_1}$, $\mathfrak{X}_{j_2}$, constructed by fixing the codewords $\mathbf{c^{j_1}}$ and $\mathbf{c^{j_2}}$ accordingly, contain a singleton and the first claim of the lemma is proved.

Since the size of the code is independent to the codeword $\mathbf{c^j}$ we choose to fix, and consequently to the corresponding family $\mathfrak{X}_j$, using (a) and Corollary 6.3.5, we obtain the desired bound

$$m \leq \binom{\ell - 1}{\lfloor \frac{\ell-1}{2} \rfloor} + 2.$$

□

The following theorem, proves the main result of the chapter, which is an improvement of the upper bound on the size of a 2-wFP code, given by Stinson and Wei in [50]:

Theorem 6.3.7 (Theorem 5.2, [50])**.** *Let C be a 2-wFP code of length ℓ and size m. Then*

$$m \leq \binom{\ell}{\lfloor \frac{\ell}{2} \rfloor} + 1. \tag{6.3.4}$$

Proof. Let $\mathbf{c^j}$ be a fixed codeword of the 2-wFP code C, and $\mathfrak{X}_j$ the corresponding family, which according to Lemma 6.3.2 is Sperner. Since the sets in $\mathfrak{X}_j$ denote the positions where all codewords agree with $\mathbf{c}^j$, the Sperner family is defined over the ground set $\{1, \ldots, \ell\}$. An upper bound of a Sperner family is given by Theorem 2.1.2, hence

$$|\mathfrak{X}_j| \leq \binom{\ell}{\lfloor \frac{\ell}{2} \rfloor}.$$

Combining this result with (6.3.2), we obtain the desired bound. □

Theorem 6.3.8. *Let $C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\}$ be a 2-wFP code of length ℓ and size m over the alphabet Q. Then*

$$m \leq \binom{\ell}{\lfloor \frac{\ell}{2} \rfloor} - \frac{\ell}{2} + 1. \tag{6.3.5}$$

Proof. Let $\mathbf{c^j}$ be a fixed codeword and $\mathfrak{X}_j$ the corresponding family as defined in (6.3.1). By Lemma 6.3.2, $\mathfrak{X}_j$ is Sperner family. Then, equation (6.3.2) indicates that in order to improve the size of the code C, it is sufficient to improve the size of $\mathfrak{X}_j$. The proof is divided into two cases with regard to the length of the code.

First, we examine the case where ℓ is even. In this case, the bound on the size of the code is given by Theorem 2.2.8 (Schonheim [44]), as its conditions are precisely the properties that family $\mathfrak{X}_j$ possesses: all sets are incomparable and the union of any pair does not cover the ground set $\{1, \ldots, \ell\}$. Hence, we have

$$|\mathfrak{X}_j| \leq \binom{\ell}{\frac{\ell}{2} - 1}$$

and using (6.3.2) we obtain the upper bound for m:

$$m \leq \binom{\ell}{\frac{\ell}{2} - 1} + 1. \tag{6.3.6}$$

Clearly, the above inequality gives a better bound than the bound (6.3.5) of the claim.

Next, we consider the case where ℓ is odd. Let $\mathfrak{X}_j = \mathcal{A} \cup \mathcal{B}$, where

$\mathcal{A} = \{A \in \mathfrak{X}_j \,:\, |A| \leq \frac{\ell-1}{2}\}$,
$\mathcal{B} = \{B \in \mathfrak{X}_j \,:\, |B| \geq \frac{\ell+1}{2}\}$.

Furthermore, we divide the family $\mathcal{A} = \mathcal{A}^- \cup \mathcal{A}^0$, where

$\mathcal{A}^- = \{A^- \in \mathfrak{X}_j \,:\, |A^-| < \frac{\ell-1}{2}\}$,
$\mathcal{A}^0 = \{A^0 \in \mathfrak{X}_j \,:\, |A^0| = \frac{\ell-1}{2}\}$,

and similarly the family $\mathcal{B} = \mathcal{B}^0 \cup \mathcal{B}^-$, where

$\mathcal{B}^0 = \{B^0 \in \mathfrak{X}_j \,:\, |B^0| = \frac{\ell+1}{2}\}$,
$\mathcal{B}^- = \{B^- \in \mathfrak{X}_j \,:\, |B^-| > \frac{\ell+1}{2}\}$,

We examine different cases according to the sizes of the sets that belong to the family $\mathfrak{X}_j$.

Case 1: $\mathfrak{X}_j = \mathcal{A} = \mathcal{A}^- \cup \mathcal{A}^0$.

First, we observe that the union of any pair of sets from $\mathcal{A}^-$ has size less than $\ell - 3$. To see this, let $A_1, A_2 \in \mathcal{A}^-$. Then, $|A_1 \cap A_2| \geq 0$ and hence

$$|A_1 \cup A_2| \leq 2\frac{\ell-3}{2} - 0 = \ell - 3.$$

Similarly, for every $A^0 \in \mathcal{A}^0$ and $A \in \mathcal{A}^-$, the union $A^0 \cup A$ does not cover the ground set $\{1, \ldots, \ell\}$:

$$|A^0 \cup A| \leq \frac{\ell-3}{2} + \frac{\ell-1}{2} - 0 = \ell - 2.$$

If the family $\mathcal{A}^0$ is not intersecting, then an upper bound on the size of C is obtained by Lemma 6.3.6:

$$m \leq \binom{\ell-1}{\frac{\ell-1}{2}} + 2.$$

It is easy to see that the above bound is smaller than the one given by (6.3.5).

$$\begin{aligned}
\binom{\ell-1}{\frac{\ell-1}{2}} + 2 &\leq \binom{\ell}{\frac{\ell-1}{2}} - \frac{\ell}{2} + 1 \\
\Leftrightarrow \quad \frac{\ell+1}{2\ell}\binom{\ell}{\frac{\ell-1}{2}} + 1 + \frac{\ell}{2} &\leq \binom{\ell}{\frac{\ell-1}{2}} \\
\Leftrightarrow \quad \frac{\ell+2}{2} &\leq \frac{\ell-1}{2\ell}\binom{\ell}{\frac{\ell-1}{2}} \\
\Leftrightarrow \quad \frac{\ell(\ell+2)}{\ell-1} &\leq \binom{\ell}{\frac{\ell-1}{2}},
\end{aligned}$$

where the last inequality is true for all $\ell \geq 5$.

Thus, we can assume that $\mathcal{A}^0$ is intersecting. For the family $\mathcal{A}$, with $\mathcal{A}^0$ being intersecting, define $\overline{\mathcal{A}}$ to be its complementary family:

$$\overline{\mathcal{A}} = \{\overline{A} \subseteq \{1,\dots,\ell\} \,:\, \overline{A} = \{1,\dots,\ell\} \setminus A,\, A \in \mathcal{A}\} = \overline{\mathcal{A}^-} \cup \overline{\mathcal{A}^0}.$$

Then, $\overline{\mathcal{A}}$ is 2-intersecting, since for all $\overline{A}_1, \overline{A}_2 \in \overline{\mathcal{A}}$ we have

$$|\overline{A}_1 \cap \overline{A}_2| = \ell - |A_1 \cup A_2| \geq \ell - (\ell - 2) = 2$$

and using Milner's bound from Theorem 2.2.6, we obtain the following

$$|\overline{\mathcal{A}}| \leq \binom{\ell}{\frac{\ell-1}{2} - 1}.$$

The families $\mathcal{A}$ and $\overline{\mathcal{A}}$ share the same cardinality, thus we can use the above result and determine a bound on the size of C:

$$m = |\mathfrak{X}_j| + 1 = |\mathcal{A}| + 1 = |\overline{\mathcal{A}}| \leq \binom{\ell}{\frac{\ell-1}{2} - 1} + 1,$$

which can be proved that is smaller than the bound of (6.3.5):

$$\begin{aligned}
\binom{\ell}{\frac{\ell-1}{2} - 1} + 1 &\leq \binom{\ell}{\frac{\ell-1}{2}} - \frac{\ell}{2} + 1 \\
\Leftrightarrow \quad (\frac{\ell-1}{\ell+3})\binom{\ell}{\frac{\ell-1}{2}} &\leq \binom{\ell}{\frac{\ell-1}{2}} - \frac{\ell}{2} \\
\Leftrightarrow \quad \frac{\ell}{2} &\leq (\frac{2}{\ell+3})\binom{\ell}{\frac{\ell-1}{2}} \\
\Leftrightarrow \quad \frac{\ell(\ell+3)}{4} &\leq \binom{\ell}{\frac{\ell-1}{2}}.
\end{aligned}$$

The last inequality holds for $\ell \geq 5$, as the order of the binomial coefficient exceeds the order of ℓ^2.

Case 2: $\mathfrak{X}_j = \mathcal{B} = \mathcal{B}^- \cup \mathcal{B}^0$.

By hypothesis, the code C is 2-wFP, which according to Proposition 6.3.3 implies that the Sperner family $\mathfrak{X}_j$ is also non 2-covering.

Similarly to the previous case, we examine the complementary family of $\mathcal{B}$:

$$\overline{\mathcal{B}} = \{\overline{B} \subseteq \{1,\dots,\ell\} \,:\, \overline{B} = \{1,\dots,\ell\} \setminus B,\, B \in \mathcal{B}\} = \overline{\mathcal{B}^-} \cup \overline{\mathcal{B}^0}.$$

Notice, that $\overline{\mathcal{B}}$ is intersecting, as for all $\overline{B}_1, \overline{B}_2 \in \overline{\mathcal{B}}$ we have

$$|\overline{B}_1 \cap \overline{B}_2| = \ell - |B_1 \cup B_2| \geq \ell - (\ell - 1) = 1,$$

where $|B_1 \cup B_2| \leq \ell - 1$ because the family $\mathfrak{X}_j$, and subsequently $\mathcal{B}$, is non 2-covering. Now, we can apply the Erős-Ko-Rado Theorem 2.2.4 on $\overline{\mathcal{B}}$, since it consists of sets of size less than $\frac{\ell-1}{2}$, and get

$$|\overline{\mathcal{B}}| \leq \binom{\ell - 1}{\frac{\ell-1}{2} - 1}$$

which in combination to (6.3.2) yields

$$m = |\mathfrak{X}_j| + 1 = |\mathcal{B}| + 1 = |\overline{\mathcal{B}}| + 1 \leq \binom{\ell - 1}{\frac{\ell-1}{2} - 1} + 1.$$

and we have already seen that this bound is an improvement to the bound given in (6.3.5).

Case 3: $\mathfrak{X}_j = \mathcal{A}^0 \cup \mathcal{B}^0$.

As previously proved, if the family $\mathcal{A}^0$ in not intersecting, then the bound on the size of the code is given by Lemma 6.3.6. Hence, we examine the case where $\mathcal{A}^0$ is intersecting. Moreover, the complementary family $\overline{\mathcal{B}^0}$ is also intersecting, as a result of $\mathcal{B}^0$ being non 2-covering. We next prove, that $\mathcal{A}^0 \cup \overline{\mathcal{B}^0}$ is intersecting, as well. Assume for a contradiction, that there exist sets $A \in \mathcal{A}^0$ and $\overline{B} \in \overline{\mathcal{B}^0}$, such that $A \cap \overline{B} = \emptyset$. This implies that $A \subseteq B$, which violates the Sperner property of $\mathfrak{X}_j$. Hence, $\mathcal{A}^0 \cup \overline{\mathcal{B}^0}$ is indeed intersecting, and consists of sets having size $\frac{\ell-1}{2}$. Applying once more the Erős-Ko-Rado Theorem 2.2.4 on $\mathfrak{X}_j$, we derive the following bound for C:

$$m = |\mathfrak{X}_j| + 1 = |\mathcal{A}^0 \cup \mathcal{B}^0| + 1 = |\mathcal{A}^0 \cup \overline{\mathcal{B}^0}| + 1 \leq \binom{\ell - 1}{\frac{\ell-1}{2} - 1} + 1,$$

which is smaller than the bound (6.3.5).

The same bound, regarding non 2-covering intersecting Sperner families that contain sets of arbitrary size, was also proved by Katona [32], under the name of *qualitatively independent sets*.

Table 6.1: Upper bounds on the size m of a 2-wFP code of odd length ℓ.

The family $\mathfrak{X}_j$	**Upper Bound**	**Characteristics**
$\mathcal{A}^- \cup \mathcal{A}^0$	$\binom{\ell}{\frac{\ell-1}{2}-1}+1$	$\mathcal{A}^0$ intersecting
	$\binom{\ell-1}{\frac{\ell-1}{2}}+2$	$\mathcal{A}^0$ not intersecting
$\mathcal{B}^- \cup \mathcal{B}^0$	$\binom{\ell-1}{\frac{\ell-1}{2}-1}+1$	–
$\mathcal{A}^0 \cup \mathcal{B}^0$	$\binom{\ell-1}{\frac{\ell-1}{2}-1}+1$	$\mathcal{A}^0$ intersecting
	$\binom{\ell-1}{\frac{\ell-1}{2}}+2$	$\mathcal{A}^0$ not intersecting
$\mathcal{A} \cup \mathcal{B}$ and the difference between the size of the smallest and the largest set is greater than 1	$\binom{\ell}{\frac{\ell-1}{2}}-\frac{\ell}{2}+1$	–

Case 4: $\mathfrak{X}_j = \mathcal{A} \cup \mathcal{B}$, with $\mathcal{A} \neq \emptyset$, $\mathcal{B} \neq \emptyset$ and the difference between the size of the smallest and the largest set in $\mathfrak{X}_j$, is greater than 1.

The bound on the size of C in this case follows directly from Proposition 2.1.4(a):

$$m = |\mathfrak{X}_j| + 1 \leq \binom{\ell}{\frac{\ell-1}{2}} - \frac{\ell}{2} + 1.$$

Table 6.1 summarises the results derived from each case, when ℓ is odd.

□

We conclude this section with a discussion on the results on wide-sense 2-frameproof codes. Clearly, in the case where the length ℓ is even, the

leading term of the upper bound on the size of 2-wFP codes in this thesis improves the previous known upper bound (Theorem 6.3.7) by a factor of $\ell/(\ell+2)$. On the other hand, when ℓ is odd the improvement is better, but still the difference between this bound and the bound of [50] tends to infinity as ℓ grows. However, the intermediate results that are presented in Table 6.1 show that when the non 2-covering Sperner family is of particular structure, the bound is significantly improved. For example, the family $\mathcal{A}_0$ being non intersecting, leads to an improvement by a factor that changes the order of the initial upper bound of Stinson and Wei [50]. Furthermore, the particular form of the non 2-covering Sperner family favours the construction of the code, since the relations amongst the codewords follow the sets in the family, and hence are limited. The case which results in a minor improvement is when the non 2-covering Sperner family contains sets of size greater than $(\ell+1)/2$ and less than $(\ell-1)/2$. Intuitively, when a Sperner family consists of sets of distant sizes, its size is likely to be much smaller than the upper bound (6.3.5), because a small set is included in many large sets. Nevertheless, a rigorous and mathematical argument that will prove the intuition is yet to be found.

6.4 2-wFP Codes of Length 5

The study of wide-sense 2-frameproof codes is concluded with the examination of length $\ell = 5$. Even though this case belongs to the category of 2-wFP codes of small length, it is presented here because its analysis requires some of the results on the structure of these codes, that were proved in the previous section. The result of this case shows that the maximal size of 2-wFP codes of length $\ell = 5$ can be attained when using an alphabet with more than three letters, while the binary and ternary case result in smaller maximal size codes.

Following the notation defined previously, the coincidence function $I(j, j')$

indicates the positions where codewords $\mathbf{c^j}$ and $\mathbf{c^{j'}}$ agree. For a fixed codeword $\mathbf{c^j} \in C = \{\mathbf{c^1}, \ldots, \mathbf{c^m}\}$, we denote by $\mathfrak{X}_j$ the family of such sets $I(j, j')$, where $j' \in \{1, \ldots, m\} \setminus \{j\}$. Lastly, $d(\mathbf{c^j}, \mathbf{c^{j'}})$ denotes the distance between codewords $\mathbf{c^j}$ and $\mathbf{c^{j'}}$ and $d(C)$ the minimum distance of the code.

Example 6.4.1. Codes C_1, C_2 are examples of 2-wFP codes of $\ell = 5$ over an alphabet of size 2 and 4, respectively.

$C_1 = \{10000, 01000, 00100, 00010, 00001, 11111\}$,

$C_2 = \{01210, 01301, 10310, 10201, 11000, 00100, 00011, 11111\}$.

Clearly, C_1 is also a ternary 2-wFP code. The following propositions show that these sizes, $|C_1| = 6$ and $|C_2| = 8$, are the best possible for the binary, ternary and arbitrary alphabet size, respectively.

Proposition 6.4.1. *Let C be a 2-wFP code of length $\ell = 5$ and size m. Then $m \leq 8$.*

Proof. Assume for a contradiction that $m > 8$. Let $\mathbf{c^0}$ be a fixed codeword and $\mathfrak{X}_0$ be the corresponding non 2-covering Sperner family. As the length is odd, according to Table 6.1, the only case that yields a code with more than 8 codewords is the last one. This means that $\mathfrak{X}_0$ contains sets of size greater than or equal to $\frac{\ell+1}{2} = 3$ and less than or equal to $\frac{\ell-1}{2} = 2$. Also, the difference between the smallest and the largest size of the sets in $\mathfrak{X}_0$ is greater than 1. We consider different cases depending on the minimum distance of C. Without loss of generality, if $d(C) = d$ then there exists a codeword $\mathbf{c^j}$ such that $d(\mathbf{c^0}, \mathbf{c^j}) = d$. According to Lemma 6.1.2 the cases that we need to examine are when $d(C) = 1, 2, 3, 4$. When $d(C) = 1$ or $d(C) = 4$, Propositions 6.1.4 and 6.1.5 respectively, show that the size of C is at most $\ell + 1 = 6$ and thus lead to a contradiction.

Case 1: $d(C) = 3$

In this case, all sets in $\mathfrak{X}_j$ have size at most 2, which is a contradiction, since $\mathfrak{X}_0$ must also contain sets of size greater than or equal to 3.

Case 2: $d(C) = 2$

The family $\mathfrak{X}_0$ consists of sets that have size at most 3 and there exists a codeword $\mathbf{c^j}$ that agrees with $\mathbf{c^0}$ in 3 positions, that is $|I(0,j)| = 3$. Let $I(0,j') \in \mathfrak{X}_0$ be the smallest set in $\mathfrak{X}_0$, for some $\mathbf{c^{j'}} \in C$. Then by assumption, its size is either 0 or 1. If $|I(0,j')| = 0$, then we have reached a contradiction, as by the property of 2-wFP codes (Proposition 6.1.1) $\mathbf{c^0}$ and $\mathbf{c^1}$ must agree in at least one position. Hence, $|I(0,j)| = 1$, which means that $\mathfrak{X}_0$ contains a singleton. This leads again to a contradiction, since in this case Corollary 6.3.5 indicates that the code has size at most $\binom{4}{2} + 2 = 8$.

As the only case that would produce a 2-wFP code of length $\ell = 5$ and size $m > 8$ leads to a contradiction, we conclude that the size of such code is at most 8. □

Next we examine the case of a binary 2-wFP code. Furthermore, as example 6.4.1 shows that the maximal 2-wFP code requires an alphabet of size at least 4, a ternary 2-wFP codes must also be studied. Before presenting the results on the binary and the ternary case, we prove a useful lemma.

Lemma 6.4.2. *Let C be a 2-wFP code of length $\ell = 5$ and size m. Let $\mathbf{c^0} \in C$ be a fixed codeword and $\mathfrak{X}_0$ be the corresponding non 2-covering Sperner family. If $\mathfrak{X}_0$ does not contain a singleton, then $m \leq 6$.*

Proof. Assume for a contradiction that $m > 6$. This implies that $\mathfrak{X}_0$ has size at least 6. Recall that $\mathcal{A}^0$ denotes the subfamily of $\mathfrak{X}_0$ that consists of sets of size $\frac{\ell-1}{2} = 2$. In Table 6.1, and according to Lemma 6.3.6 the cases where $\mathcal{A}^0$ is not intersecting, imply the existence of a family different from $\mathfrak{X}_0$ which contains a singleton. Hence, the only case where the family $\mathfrak{X}_0$ does not contain a singleton and can lead to a code of size 6 or more, is the last one. That is, when $\mathfrak{X}_0$ contains sets of size greater than 2 and less than 3, and the difference between the size of the largest and the smallest set is at

least 2. Since the ground set $\{1,2,3,4,5\}$ is of size 5, if $\mathfrak{X}_0$ has a 5-set, then $|\mathfrak{X}_0| = |\{1,2,3,4,5\}| = 1$ and thus, $m = 2$, a contradiction. If $\mathfrak{X}_0$ contains a 4-set, say $|I(0,1)| = 4$, then we have that $d(\mathbf{c^0}, \mathbf{c^1}) = 1$. According to Lemma 6.1.2, the minimum distance of a 2-wFP code is at least 1, therefore we have $d(C) = 1$ and by Proposition 6.1.4, $m \leq 2$, again a contradiction. The last case, where $\mathfrak{X}_0$ has a 3-set, also leads to a contradiction, as it implies that the smallest set in $\mathfrak{X}_0$ must be a singleton. □

Proposition 6.4.3. *Let C be a ternary 2-wFP code of length $\ell = 5$ and size m. Then $m \leq 6$.*

Proof. Assume for a contradiction that C contains more than 6 codewords, that is $m = 7$ or $m = 8$. If there exists a ternary 2-wFP code of size 8, then by deleting one codeword we would have a code of size $m = 7$. Hence, it is sufficient to investigate only the case where $m = 7$.

Let $\mathfrak{X}_0$ be the non 2-covering Sperner family generated by the fixed codeword $\mathbf{c^0}$. According to the previous lemma, the case that could lead to a code of size greater than 6, is when $\mathfrak{X}_0$ contains a singleton. Without loss of generality, let $\{1\} \in \mathfrak{X}_0$ be the singleton. Since the family $\mathfrak{X}_0$ is Sperner, the subfamily $\hat{\mathfrak{X}}_0 = \mathfrak{X}_0 \setminus \{1\}$ must also be Sperner and in order for C to have size 7, $\hat{\mathfrak{X}}_0$ must have size 5. This is derived from equation (6.3.2):

$$m = |\mathfrak{X}_0| + 1 = |\hat{\mathfrak{X}}_0| + 2.$$

Moreover, since none of the sets in $\hat{\mathfrak{X}}_0$ can contain the letter '1', as this would violate the Sperner property of $\mathfrak{X}_0$, the ground set for this Sperner subfamily is $\{2,3,4,5\}$. Let U denote the size of the largest set in $\hat{\mathfrak{X}}_0$ and L the size of the smallest. Then, according to Proposition 2.1.4, the upper bound on the size of $|\hat{\mathfrak{X}}_0|$ for all possible values of the difference $U - L$, is the following:

- ⋄ if $U - L = 0$, then $m \leq \binom{4}{2} - 0(\frac{4}{2}) = 6$
- ⋄ if $U - L = 1$, then $m \leq \binom{4}{2} - 1(\frac{4}{2}) = 4$

Table 6.2: The Sperner family $\mathfrak{X}_0$ and the corresponding codewords.

$\mathfrak{X}_0$	$\mathbf{C_0}$	
	$\mathbf{c^0}$	11111
$\{2,3\}$	$\mathbf{c^1}$	_11__
$\{2,4\}$	$\mathbf{c^2}$	_1_1_
$\{2,5\}$	$\mathbf{c^3}$	_1__1
$\{3,4\}$	$\mathbf{c^4}$	__11_
$\{3,5\}$	$\mathbf{c^5}$	__1_1
$\{1\}$	$\mathbf{c^6}$	1____

- ⋄ if $U - L = 2$, then $m \leq \binom{4}{2} - 2(\frac{4}{2}) = 2$
- ⋄ if $U - L = 3$, then $m \leq \binom{4}{2} - 3(\frac{4}{2}) = 0$

From the above, it is clear that the only case we need to consider is when $U - L = 0$, or in other words, when $\hat{\mathfrak{X}}_0$ contains sets of the same size. It is easy to see that if $\hat{\mathfrak{X}}_0$ consists only of singletons, or 3-sets, or 4-sets, then its size will not exceed 4. Hence, $\hat{\mathfrak{X}}_0$ contains only 2-sets. Without loss of generality, assume that $\hat{\mathfrak{X}}_0 = \{\{2,3\},\{2,4\},\{2,5\},\{3,4\},\{3,5\}\}$. Translating this into codewords, we obtain the code of Table 6.2.

Notice, that reuse of letter '1' would violate the Sperner property of $\mathfrak{X}_0$, thus, each one of remaining positions can be filled using the letter '0' or '2'. For the codewords $\mathbf{c^2}$ and $\mathbf{c^5}$, we have that $I(0,2) = \{2,4\}$ and $I(0,5) = \{3,5\}$. The intersection of these sets is empty, thus the transitivity of the coincidence function implies that $I(2,5)$ cannot contain any of the positions 2, 3, 4 or 5. As by the property of 2-wFP codes (Proposition 6.1.1) any pair of codewords must agree in at least one position, we have that $I(2,5) = \{1\}$. Without loss of generality, let '0' be their common letter. Furthermore, since the first is the only position that $\mathbf{c^2}$ and $\mathbf{c^5}$ agree, by the property of 2-wFP codes the letter '0' cannot be used to fill the first position of any other codeword.

Table 6.3: The Sperner family $\mathfrak{X}_0$ and the corresponding ternary code.

$\mathfrak{X}_0$	$\mathbf{C_0}$	
	$\mathbf{c^0}$	11111
$\{2,3\}$	$\mathbf{c^1}$	_11__
$\{2,4\}$	$\mathbf{c^2}$	01_1_
$\{2,5\}$	$\mathbf{c^3}$	21__1
$\{3,4\}$	$\mathbf{c^4}$	2_11_
$\{3,5\}$	$\mathbf{c^5}$	0_1_1
$\{1\}$	$\mathbf{c^6}$	1____

Similar situation also holds for $\mathbf{c^3}$ and $\mathbf{c^4}$:

$$I(0,3) \cap I(0,4) = \{2,5\} \cap \{3,4\} = \emptyset$$

and using the same argument as previously, we obtain $I(3,4) = \{1\}$. Additionally, neither '1' nor '0' can be their common letter, as the former would violate the Sperner property of $\mathfrak{X}_0$, and the property of the 2-wFP codes, the latter. Hence, the first position of $\mathbf{c^3}$ and $\mathbf{c^4}$ is filled with the letter '2'. The code in Table 6.3 summarises these results.

As already mentioned, in order to retain the property of the 2-wFP code, only the pair $(\mathbf{c^2}, \mathbf{c^5})$ can share the letter '0' in the first position, while letter '2' covers the same position of the pair $(\mathbf{c^3}, \mathbf{c^4})$ and this pair alone. Since the alphabet consists only of three letters, there is no letter left to fill the first position of $\mathbf{c^1}$, contradicting the assumption that there exists a ternary 2-wFP code of length $\ell = 5$ and size $m > 6$. □

Regarding a 2-wFP code over the binary alphabet, the following proposition is easily derived from the previous, in combination with example 6.4.1.

Proposition 6.4.4. *Let C be a binary 2-wFP code of length $\ell = 5$ and size m. Then $m \leq 6$.*

Proof. According to the previous proposition, when the alphabet size is 3, then the maximal 2-wFP code of length $\ell = 5$ has size 6. This implies, that it is not possible to obtain a 2-wFP code of larger size over an alphabet with fewer letters. Hence, in the binary case such a code contains at most 6 codewords. Code C_1 from example 6.4.1, shows that there exists a binary 2-wFP code of length $\ell = 5$ and size 6, and the proof is concluded. □

Part II

Honeycomb Arrays

Chapter 7

Honeycomb Arrays

Honeycomb arrays are combinatorial objects that emerged from the study of Costas arrays. Costas arrays were first introduced in an initially classified report on signal processing, by J.P. Costas [19] and since then their study is still evolving. Due to this close relation between the two objects, it is necessary to examine Costas arrays as well.

The first section of this chapter presents the definitions of Costas and honeycomb arrays and studies the way these combinatorial objects are connected. A more detailed review on Costas arrays is presented in the second section, and includes constructions and computational results regarding their enumeration. Honeycomb arrays are covered in section three, which is divided into two parts. The first part proves an original result (Theorem 7.3.1) related to the construction of honeycomb arrays and shows that the only way of constructing them is by using Costas arrays. The enumeration of honeycomb arrays was first initiated in 1984 by Golomb and Taylor [29]. The second part continues this enumeration by filling the gaps and updating the list of known honeycomb arrays. For a clearer view, a classification of the known honeycomb arrays is also presented in the third section, together with the corresponding Costas arrays that they generated from. The chapter concludes with some remarks on the symmetric properties of honeycomb arrays.

7.1 From Rooks to Semi-Queens

Costas arrays are $n \times n$ matrices, on which we have placed n dots that satisfy the following two conditions: we allow exactly one dot to be placed in each row and in each column and we also require that the vector differences between all pairs of dots, thought of as vectors, are different. Due to the similarity to the move of the Rook on the chess board, the first condition is sometimes called the non attacking Rook property.

Example 7.1.1. A simple example of Costas array is the 3×3 matrix with the following dot configuration:

Figure 7.1: A Costas array.

It is easy to check that the six vector differences between all pairs of dots are distinct.

Definition 7.1.1. A Costas array of order n is a configuration of n dots placed on a $n \times n$ square grid in such a way that the following conditions are satisfied:

(a) in each row and in each column of the grid exactly one dot occurs and

(b) all $n(n-1)$ pairs of dots have distinct vector differences.

The hexagonal analogue of Costas array is called a honeycomb array. A simple example of a honeycomb array with 3 dots is shown in Figure 7.2.

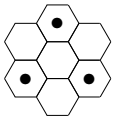


Figure 7.2: A honeycomb array.

As honeycomb arrays exist in the hexagonal environment, it is essential to describe first the hexagonal board. Define the hexagonal grid to be the lattice which is generated by the following two vectors:

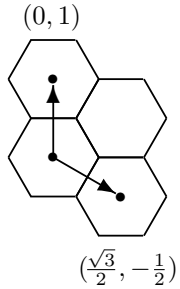


Figure 7.3: The hexagonal grid.

In contrast to the square, the directions defined by the hexagonal grid are three. As Figure 7.4 shows, two follow the diagonals, while the third direction is determined by the columns.

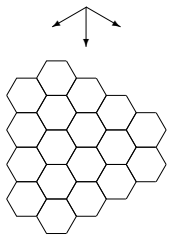


Figure 7.4: Hexagonal region.

Definition 7.1.2. A honeycomb array of order n is a configuration of n dots placed on a hexagonal grid in such a way that the following conditions are satisfied:

(a) in each diagonal (in both directions) and in each column of the grid at most one dot occurs,

(b) all n dots lie in consecutive diagonals (in both directions) and in consecutive columns,

(c) all $n(n-1)$ pairs of dots have distinct vector differences.

Honeycomb arrays appeared in 1997, in a paper by Golomb and Taylor [29] where they were derived from Costas arrays, using a shear- compression transformation. Let C be a Costas array and define the main diagonal to be the diagonal that runs from the top right corner to the bottom left. Using the similarity between the dot configuration of the Costas array and the moves of the chess pieces, an attacking Queen is a configuration where one dot can attack its row, column or its diagonals. Following the terminology of Golomb and Taylor, an attacking semi-Queen can attack its row, column and only the diagonal parallel to the main diagonal. If we apply shear-compression to the Costas array with non attacking semi-Queen dot configuration, then Figure 7.5 shows that the result is the hexagonal analogue of that particular Costas array or in other words, a honeycomb array. From the same figure it is clear that while the columns of the Costas array are not affected by the transformation, the rows become diagonals.

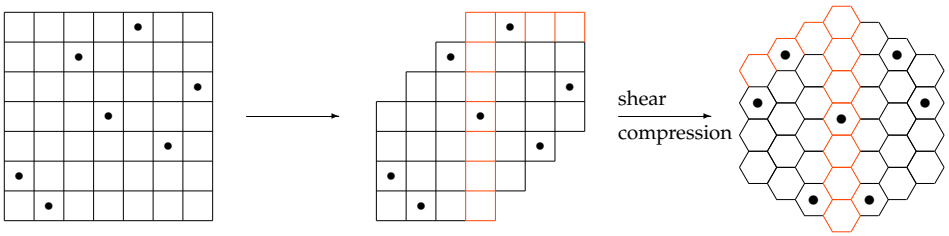


Figure 7.5: Golomb and Taylor construction of honeycomb arrays from Costas arrays.

Translating the properties of the Costas array into hexagonal terms, after the shear-compression each diagonal (in both directions) and each column, contains exactly one dot. Additionally, the property of the Costas arrays of the distinct vector differences between all pairs of dots, is preserved.

Let $\xi : \mathbb{R}^2 \to \mathbb{R}^2$ be the linear transformation which converts the square into the hexagonal grid:

$$\xi(x, y) = (\frac{\sqrt{3}}{2}x, -\frac{1}{2}x + y).$$

Clearly, the inverse of ξ, which maps (x, y) to $(\frac{2}{\sqrt{3}}x, \frac{1}{\sqrt{3}}x + y)$, transforms the

hexagonal into the square lattice. Define a *point* of the hexagonal lattice to refer to the centre of the hexagonal cell. The *hexagonal distance* between two points x and y is the smallest r, such that there exists a path with $r+1$ points

$$x = p_1, p_2, \ldots, p_{r+1} = y,$$

where p_i and p_{i+1} are adjacent points in the hexagonal grid. Clearly, the distance of adjacent points on the hexagonal lattice is one and thus, each point is surrounded by 6 neighbours. Define the *hexagonal sphere* of radius r, also known as *Lee sphere*, to be the area on the hexagonal lattice that consists of a fixed point, the centre, together with all points that are at distance r or less from the centre. Using these definitions we next introduce the notion of a honeycomb array of radius r.

Definition 7.1.3. A honeycomb array of radius r is a configuration of $n = 2r + 1$ dots on the hexagonal grid with

(a) every dot being at distance at most r from a fixed hexagon (the centre),

(b) exactly one dot in each column and each diagonal (in both directions) (non attacking bee-Rooks),

(c) distinct vector differences between all pairs of dots.

In [13] Blackburn, Panoui, Paterson and Stinson proved the following theorem which shows that honeycomb arrays are actually honeycomb arrays of radius r having an odd number of dots.

Theorem 7.1.4 (Corollary 2, [13])**.** *Any honeycomb array is a honeycomb array of radius r, for some integer r. In particular, a honeycomb array must consist of an odd number of dots.*

Proof. Define a hexagonal permutation π to be a collection of $n \times n$ dots which satisfy (a) and (b) of Definition 7.1.2. Clearly, any honeycomb array is a hexagonal permutation. Moreover, since a honeycomb array of radius

r is contained in a Lee sphere of the same radius, the proof of the theorem is reduced to proving the following claim: for any hexagonal permutation π with n dots, n is odd and the dots in π are contained in a Lee sphere of radius $\frac{n-1}{2}$.

By slightly abusing the notation, let $\xi^{-1}(\pi)$ denote the image of π in the square grid. Also, for non negative integers i, m with $1 \leq i \leq m-1$, let $S_i(m)$ denote the region depicted in Figure 7.6. It is easy to prove that all dots in ξ^{-1} are included in a region of the form $S_i(n)$, for some $1 \leq i \leq n-1$ and they form a non attacking semi-Queen configuration (Lemma 3, [13]).

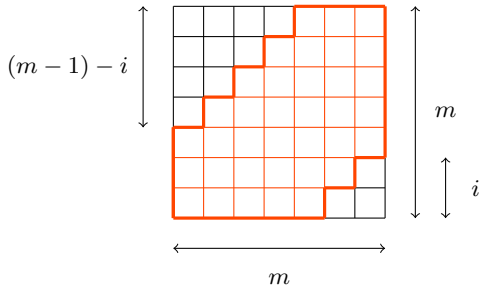


Figure 7.6: The region $S_i(m)$.

If $i = \frac{n-1}{2}$, which means that n is an odd number, the claim follows and consequently the theorem is proved for this case. Assume for a contradiction that $i \neq \frac{n-1}{2}$. By reflecting π vertically in the hexagonal grid, we obtain a permutation π' such that the dots in $\xi^{-1}(\pi')$ are included in a region of the form $S_{(n-1)-i}(n)$. Hence, we can assume that $i < \frac{n-1}{2}$, by replacing π with π' if required.

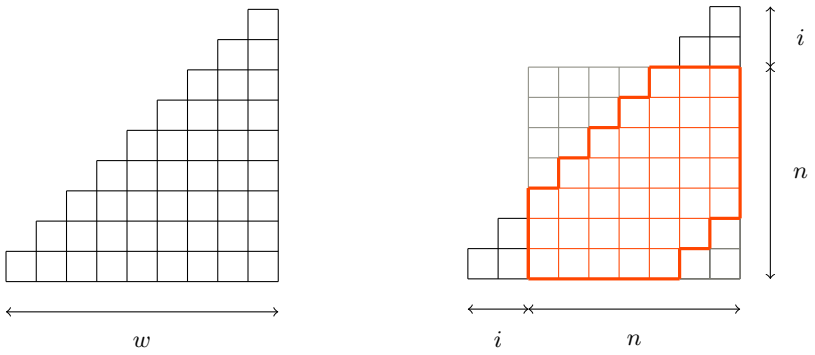
Consider the triangular board of width $w = n + i$ (Figure 7.7(a)) containing $S_i(n)$ (Figure 7.7(b)). According to a result in [39] by Nivasch and Lev and in [56] (P252 and R252) by Vaderlind, Guy and Larson the maximum number of dots forming a non attacking semi-Queen configuration that can be placed in a triangular board of width w is $\frac{2w+1}{3}$. Applying this to the triangular board of width $n + i$, we obtain

$$\frac{2(n+i)+1}{3}.$$

Since $0 \leq i \leq n-1$ we have that

$$\frac{2(n+i)+1}{3} \leq \frac{2n+(n-1)+1}{3} = n$$

which is a contradiction. Thus, the only choice for i is to be equal to $\frac{n-1}{2}$.



(a) (b)

Figure 7.7: The triangular board and how is covered by the region $S_i(n)$.

□

7.2 Costas Arrays

The research of Costas arrays is primarily directed by the following two questions:

1. How many Costas arrays of order n are there?
2. How can they be constructed?

A detailed presentation of the known constructions of Costas arrays can be found in [21] by Drakakis. For completeness, these constructions are presented below. Let $\mathbb{F}_q$ be a finite field and α a primitive element of $\mathbb{F}_q$. For all $x \in \mathbb{F}_q^*$, define $\log_\alpha x$ to be the discrete logarithm of x to the base α.

Welsh construction [29] For every prime p, the Welsh construction generates Costas arrays of order $n = p-1\,(W_1)$, $n = p-2\,(W_2)$ and $n = p-3\,(W_3)$ when $\alpha = 2$:

$\mathbf{W_1}$: For all $i = 1, \ldots, p-1$ and $j = 0, \ldots, p-2$, put a dot in position (i, j) if and only if $i = \alpha^j$.

$\mathbf{W_2}$: From the Costas array obtained from the previous construction, delete the dot at $(1, 0)$, together with the corresponding row and column.

$\mathbf{W_3}$: For $\alpha = 2$, delete the two dots at $(1, 0)$ and $(2, 1)$ from the Costas array generated from W_1, together with the corresponding rows and columns.

Lempel construction [29] In this construction q is any prime power, $q = p^k$, and yields Costas arrays of order $n = q - 2\,(L_2)$ and $n = q - 3\,(L_3)$ whenever 2 is a primitive element of $\mathbb{F}_q$:

$\mathbf{L_2}$: For all $i = 1, \ldots, p-2$ and $j = 1, \ldots, p-2$, put a dot in position (i, j) if and only if $\alpha^i + \alpha^j = 1$.

$\mathbf{L_3}$: If 2 is a primitive element of $\mathbb{F}_q$, then delete the dot at $(1, 1)$ from the Costas array constructed using L_2, together with the corresponding row and column.

Taylor construction (variant of Lempel) [29] This construction, denoted by T_4, can be applied when we have that $\alpha^1 + \alpha^2 = 1$ for the primitive element α of $\mathbb{F}_q$, with $q = p^k$. In this case, by deleting the dots at $(1, 1)$ and $(2, 1)$ together with the corresponding rows and columns from the Costas array constructed using L_2, we obtain a Costas array of order $n = q - 4$.

Golomb construction [29] Let α and β be primitive elements of the field $\mathbb{F}_q$, where q is a prime power. Then, the Costas arrays constructed using this method have order $n = q-2\,(G_2)$, $n = q-3\,(G_3)$ and $n = q-4\,(G_4)$ only when $q = 2^k$ and $\alpha + \beta = 1$:

$\mathbf{G_2}$: For all $i = 1, \ldots, p-2$ and $j = 1, \ldots, p-2$, put a dot in position (i, j) if and only if $\alpha^i + \beta^j = 1$.

$\mathbf{G_3}$: If $\alpha^1+\beta^1=1$, then from the Costas array obtained from G_2, delete the dot at (1,1) together with the corresponding row and column.

$\mathbf{G_4}$: For $q=p^k$ and when $\alpha^1+\beta^1=1$ in $\mathbb{F}_q$, delete the two dots together with the corresponding rows and columns at positions $(1,1)$ and $(2,2)$, from the Costas array generated by construction G_1.

Golomb variant [29] This variant generates Costas arrays of order $n=q-4\,(G_4^*)$ and $n=q-5\,(G_5^*)$, both under some necessary conditions:

$\mathbf{G_4^*}$: When $\alpha^1+\beta^1=1$ and $\alpha^2+\beta^{-1}=1$, where α and β are primitive elements of $\mathbb{F}_q$, then delete the dot at $(1,1)$ and $(2,q-2)$ from construction G_2.

$\mathbf{G_5^*}$: This construction succeeds G_4^*, since when deleting the dots at $(1,1)$ and $(2,q-2)$, the resulting Costas array has a dot at $(q-2,2)$ which can also be deleted. This follows from the fact that if $\alpha^1+\beta^1=1$ and $\alpha^2+\beta^{-1}=1$, then also $\alpha^{-1}+\beta^2=1$.

Taylor construction (variant of Golomb) [29] Taylor variant yields constructions of Costas arrays of order $n=q-1\,(T_1)$ and $n=q\,(T_0)$. Instead of deleting dots, the new arrays are produced by adding dots in specific positions, in such a way that the properties of Costas arrays are preserved:

$\mathbf{T_1}$: When $q\neq 2^k$, then add a dot at one of the positions $(0,0)$, $(0,q-1)$, $(q-1,0)$ or $(q-1,q-1)$, such that the resulting array is Costas.

$\mathbf{T_0}$: When $q\equiv -1 \mod 6$, then add two dots at positions $(0,0)$ and $(q-1,q-1)$ or at $(0,q-1)$ and $(q-1,0)$, such that the resulting array is Costas.

Rickard construction [43] This construction is based on the periodicity properties of existing constructions. That is, a $n\times n$ Costas array is repeated vertically and horizontally (double periodicity), in order to produce an

array that contains a Costas array in every $n \times n$ window. Since Taylor [52] proved that for $n > 2$ such an array does not exist, Rickard [43] modified the above method and instead of the double repetition, the $n \times n$ Costas array is vertically repeated (single periodicity) and a row between every two arrays is left empty. A Costas array is produced, by taking a $(n+1) \times n$ window from this construction of repeated Costas arrays, and searching the correct position to place the $(n+1)^{th}$ dot in the empty row (1-Gap Augmentation method).

The symmetries of the square allow us to partition the set of Costas arrays into equivalence classes. For example, an equivalence class can be created by rotating or reflecting a Costas array. In order to avoid confusion when we enumerate the Costas arrays, it is necessary to define the following parameters:

$C(n)$: the total number of $n \times n$ Costas arrays

$c(n)$: the number of equivalence classes of $n \times n$ Costas arrays

In 2008 a group formed by Taylor, Drakakis and Rickard [53] constructed a toolbox of functions that carry out an exhaustive search for Costas arrays, which led to finding all Costas arrays up to order $n = 26$. Later, Drakakis et al. [24] completed an exhaustive search for $n = 27$. Recently, Drakakis, Iorio and Rickard [22], and Drakakis, Iorio, Rickard and Walsh [23] presented all Costas arrays of order $n = 28$ and $n = 29$, respectively. These enumeration results are presented in Table 7.1. A list of all Costas arrays up to order $n = 200$, that are constructed using the algebraic methods presented above, can be found in the database of Beard [7]. The results of the exhaustive search reveals the existence of sporadic Costas arrays, that is, arrays that do not follow any of the aforementioned algebraic constructions.

Table 7.1: The number of $n \times n$ Costas arrays found by exhaustive search.

n	1	2	3	4	5	6	7	8	9	10	11	12
$C(n)$	1	2	4	12	40	116	200	444	760	2160	4368	7852
$c(n)$	1	1	1	2	6	17	30	60	100	277	555	990

n	13	14	15	16	17	18	19	20
$C(n)$	12828	17252	19612	21104	18276	15096	10240	6464
$c(n)$	1616	2168	2467	2648	2294	1892	1283	810

n	21	22	23	24	25	26	27	28	29
$C(n)$	3536	2052	872	200	88	56	204	712	164
$c(n)$	446	259	114	25	12	8	29	89	23

7.3 Honeycomb Arrays

7.3.1 Construction of Honeycomb Arrays

The constructions of Costas arrays, and in particular the Welsh method, show that Costas arrays exists for infinitely many values of n. For honeycomb arrays however, the scenario is not as encouraging. In [11] Blackburn, Etzion, Martin and Paterson prove that the values of the radius r for which there exist honeycomb arrays, are finite. In particular, the authors show that honeycomb arrays do not exist for $r \geq 664$. Regarding their construction, one method was introduced by Golomb and Taylor in [29]. As previously mentioned, it is based on applying shear-compression to a Costas array with non attacking semi-Queens dot configuration. The lack of any other way of constructing honeycomb arrays led to a conjecture claiming that these arrays can only be constructed by Costas arrays, using shear-compression. This section proves the conjecture to be true.

Recall that the inverse of transformation ξ, as defined in Section 7.1, converts the hexagonal into the square grid.

Theorem 7.3.1. *Let H be a honeycomb array of radius r. Then $\xi^{-1}(H) = C$, where C is a Costas array of size $n = 2r + 1$.*

Proof. The claim will be proved in three steps. The first step shows that the map ξ^{-1} is a bijection and when is applied to a hexagonal sphere, the resulting shape is contained in a square. Second, we prove that in the derived configuration, the vector differences between all pairs of dots are distinct. Finally, we need to show that there is exactly one dot in each row and each column of the constructed array (non attacking Rooks).

Step 1: The map ξ^{-1}.

Since any linear transformation can be described by a matrix, in order to show that ξ^{-1} is a bijection, it only suffices to show that the determinant of its matrix is nonzero. The map ξ^{-1} corresponds to the matrix

$$\begin{pmatrix} \frac{2}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{3}} & 1 \end{pmatrix}$$

whose determinant is $2/\sqrt{3}$, thus $\xi^{-1}$is a bijection.

We continue by showing that the centres of the hexagons in the hexagonal sphere are transformed via ξ^{-1} into a subset of the square lattice. Without loss of generality, we begin with a small hexagonal sphere of radius $r = 1$, centred at (0,0). We calculate the centres of the 7 hexagons and apply ξ^{-1} to switch to the square grid (Figure 7.8):

$O = \xi^{-1}(O) = \xi^{-1}(0,0) = (0,0),$

$A' = \xi^{-1}(A) = \xi^{-1}(0,1) = (0,1),$

$B' = \xi^{-1}(B) = \xi^{-1}(\frac{\sqrt{3}}{2}, \frac{1}{2}) = (\frac{2}{\sqrt{3}}\frac{\sqrt{3}}{2}, \frac{1}{\sqrt{3}}\frac{\sqrt{3}}{2} + \frac{1}{2}) = (1,1),$

$C' = \xi^{-1}(C) = \xi^{-1}(\frac{\sqrt{3}}{2}, -\frac{1}{2}) = (\frac{2}{\sqrt{3}}\frac{\sqrt{3}}{2}, \frac{1}{\sqrt{3}}\frac{\sqrt{3}}{2} - \frac{1}{2}) = (1,0),$

$D' = \xi^{-1}(D) = \xi^{-1}(0,-1) = (0,-1),$

$E' = \xi^{-1}(E) = \xi^{-1}(-\frac{\sqrt{3}}{2}, -\frac{1}{2}) = (-\frac{2}{\sqrt{3}}\frac{\sqrt{3}}{2}, -\frac{1}{\sqrt{3}}\frac{\sqrt{3}}{2} - \frac{1}{2}) = (-1,-1),$

$F' = \xi^{-1}(F) = \xi^{-1}(-\frac{\sqrt{3}}{2}, \frac{1}{2}) = (-\frac{2}{\sqrt{3}}\frac{\sqrt{3}}{2}, -\frac{1}{\sqrt{3}}\frac{\sqrt{3}}{2} + \frac{1}{2}) = (-1,0).$

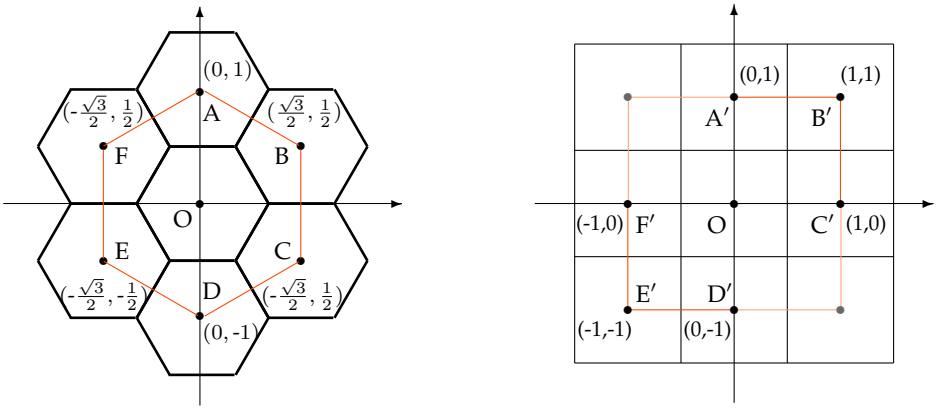


Figure 7.8: The transformation of the hexagonal lattice into the square lattice.

As Figure 7.8 shows, the resulting shape is an incomplete square, which can be completed by adding the two missing points at the top left and the bottom right corners. Let us generalize the above procedure in terms of neighbour cells. Figure 7.9 shows the neighbours of the central cell of the hexagonal array and the corresponding neighbour relation on the square array.

Figure 7.9: The analogy between the neighbours in the hexagonal and square array.

Using this relation it is easy to picture how the square array will be constructed. We prove by induction on r that a hexagonal sphere of radius r is transformed into an incomplete square of dimension $n = 2r + 1$.

Base Case

When $r = 1$, the hexagonal sphere is transformed into the square of order $n = 3$. The calculations have already been done in a previous paragraph and are presented in Figure 7.8.

Inductive Hypothesis

Assume that the hexagonal sphere of radius r is transformed into an incomplete square of order $n = 2r + 1$.

Inductive Step

Let H be a hexagonal sphere of radius $r + 1$ (Figure 7.10). By the inductive hypothesis, the hexagonal sphere of radius r (the black part of H in Figure 7.10) which is contained in H, is transformed into an incomplete square S' of order $n' = 2r + 1$ (Figure 7.10).

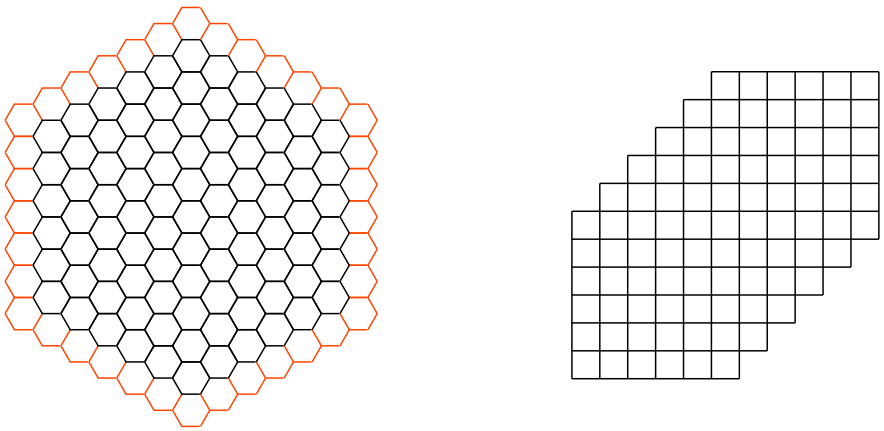


Figure 7.10: The black hexagonal sphere of order r is transformed into the incomplete square S' of order $n' = 2r + 1$.

For the remaining hexagonal cells, we use the neighbour relation and as Figure 7.11 shows, it adds one more layer of square cells to S'. Hence, the final incomplete square S, has order $n = n' + 1 = 2r + 1 + 1 = 2r + 2$, which completes the induction.

Step 2: The vector differences.

Let $H = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}$ and $C = \{\mathbf{v_1}, \ldots, \mathbf{v_n}\}$ be the sets of the centres, described by vectors, that form the honeycomb and the incomplete square array, respectively. Assume for a contradiction, that there are four vectors $\mathbf{v_i}, \mathbf{v_j}, \mathbf{v_{i'}}, \mathbf{v_{j'}} \in C$ such that $\mathbf{v_i} - \mathbf{v_j} = \mathbf{v_{i'}} - \mathbf{v_{j'}}$ and $i \neq j$,

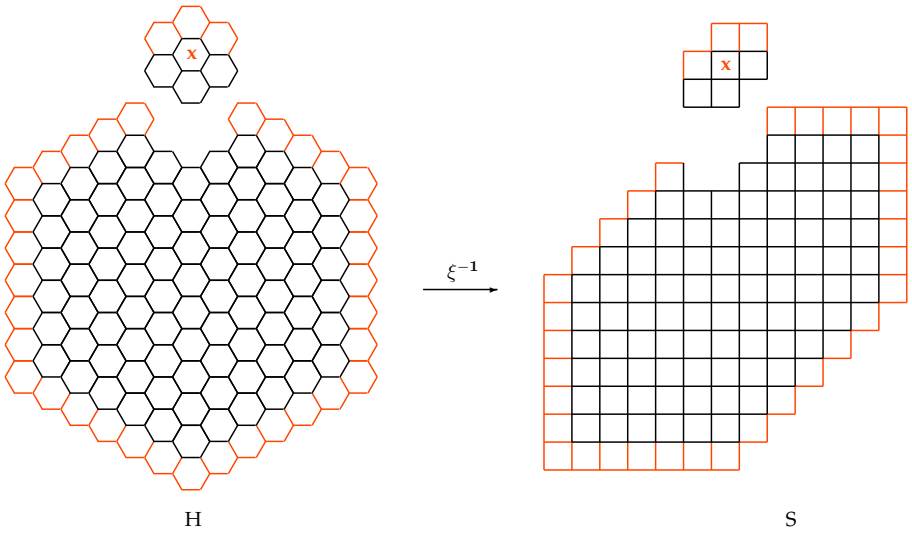


Figure 7.11: Illustration of the inductive step. The extracted shapes show the neighbour relation of the marked cell, as the hexagonal is transformed into the square lattice.

$i' \neq j'$. By construction, $\mathbf{v_k} = \xi^{-1}(\mathbf{w_k})$ for all $k = 1, \ldots, n$. Thus,

$$\xi^{-1}(\mathbf{w_i}) - \xi^{-1}(\mathbf{w_j}) = \xi^{-1}(\mathbf{w_{i'}}) - \xi^{-1}(\mathbf{w_{j'}})$$

and hence

$$\xi^{-1}((\mathbf{w_i} - \mathbf{w_j}) - (\mathbf{w_{i'}} - \mathbf{w_{j'}})) = 0,$$

which implies that

$$\mathbf{w_i} - \mathbf{w_j} = \mathbf{w_{i'}} - \mathbf{w_{j'}}.$$

Since one of the properties of the honeycomb array is the distinct vector differences, the last equality implies that $i = i'$ and $j = j'$, which is a contradiction. Thus, the linear transformation ξ^{-1} preserves the vector differences, and so if they are distinct in the honeycomb array, they are distinct in the incomplete square as well.

Step 3: Non attacking Rooks.

We prove by contradiction, that the $n \times n$ array which is constructed from the honeycomb array, has exactly one dot in each row and each column. All the possible positions of the centres in the square grid are

of the form

$$\lambda(1,0) + \mu(0,1), \quad \lambda, \mu \in \mathbb{Z}$$

and in the hexagonal grid of the form

$$\alpha(\tfrac{\sqrt{3}}{2}, \text{-}\tfrac{1}{2}) + \beta(0,1), \quad \alpha, \beta \in \mathbb{Z}$$

where $\{(1,0),(0,1)\}$ and $\{(\frac{\sqrt{3}}{2}, \text{-}\frac{1}{2}),(0,1)\}$ are the basis for the square and the hexagonal grid respectively.

Now, suppose that there exists a column in the $n \times n$ array with 2 dots in positions $\mathbf{e} = (\lambda_1, \mu)$ and $\mathbf{f} = (\lambda_2, \mu)$. We apply the map ξ to switch from the square to the hexagonal grid:

$\mathbf{e}' = \xi(\mathbf{e}) = (\frac{\sqrt{3}}{2}\lambda_1, -\frac{1}{2}\lambda_1 + \mu)$
$\mathbf{f}' = \xi(\mathbf{f}) = (\frac{\sqrt{3}}{2}\lambda_2, -\frac{1}{2}\lambda_2 + \mu)$

In order to check whether the mapped dots occur in the same diagonal in the honeycomb array, we use the vector slope. If the vector difference $\mathbf{e}' - \mathbf{f}'$ has the same slope with the basis vector $(\frac{\sqrt{3}}{2}, \text{-}\frac{1}{2})$, then the mapped dots lie in the same diagonal. Since the hexagonal grid has three directions, we also need to compare the slope of the vector difference to the slope of $(\frac{\sqrt{3}}{2}, \frac{1}{2})$ (Figure 7.12).

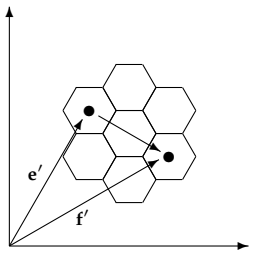


Figure 7.12: The use of slope for the determination of whether or not a diagonal in the hexagonal sphere contains two dots.

We calculate the vector difference and its slope:

$$\mathbf{e}' - \mathbf{f}' = (\tfrac{\sqrt{3}}{2}(\lambda_1 - \lambda_2), -\tfrac{1}{2}(\lambda_1 - \lambda_2)),$$
$$\lambda_{\mathbf{e}'\text{-}\mathbf{f}'} = \frac{-\frac{1}{2}(\lambda_1 - \lambda_2)}{\frac{\sqrt{3}}{2}(\lambda_1 - \lambda_2)} = -\tfrac{1}{\sqrt{3}}.$$

The slopes of vectors $(\frac{\sqrt{3}}{2}, \text{-}\frac{1}{2})$ and $(\frac{\sqrt{3}}{2}, \frac{1}{2})$ are:

$$\lambda_{(\frac{\sqrt{3}}{2}, \text{-}\frac{1}{2})} = \frac{\frac{1}{2}}{\frac{\sqrt{3}}{2}} = -\tfrac{1}{\sqrt{3}},$$
$$\lambda_{(\frac{\sqrt{3}}{2}, \frac{1}{2})} = \frac{\frac{1}{2}}{\frac{\sqrt{3}}{2}} = \tfrac{1}{\sqrt{3}}$$

The relation $\lambda_{\mathbf{e}'-\mathbf{f}'} = \lambda_{(\frac{\sqrt{3}}{2}, \text{-}\frac{1}{2})}$ indicates that there exists a diagonal in the honeycomb array with two dots, which is a contradiction.

As the transformation leaves the columns intact, if two dots occupy the same column in the square array, then the corresponding column in the honeycomb array would have two dots as well, which is a contradiction.

Hence, the $n \times n$ array that is constructed by the honeycomb array has exactly one dot in each row and column.

□

7.3.2 Computational Results

This section presents the complete list of known honeycomb arrays up to radius $r = 14$ and one array of radius $r = 22$, which is mentioned in [29] by Golomb and Taylor. The list appears in [13] for $r \leq 13$, while recent computational results on Costas arrays of order $n = 29$ [23] allowed the search for honeycomb arrays of radius $r = 14$. Similar to the case of Costas arrays, honeycomb arrays can also be partitioned into equivalence classes, according to which symmetries of the hexagon they follow. According to the notation of Golomb and Taylor

$H(r)$: denotes the total number of honeycomb arrays of radius r

$h(r)$: denotes the number of equivalence classes of honeycomb arrays of radius r.

Example 7.3.1. The honeycomb arrays in Figures 7.13 and 7.14 are the four known of radius 4 and they can be partitioned in two equivalence classes.

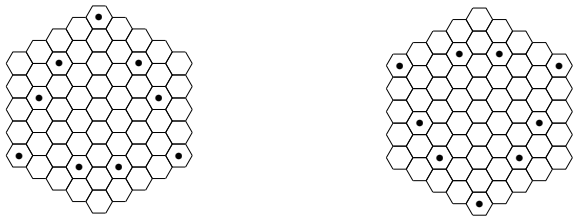


Figure 7.13: The two members of the first equivalence class, where the second honeycomb array is the vertical reflection of the first.

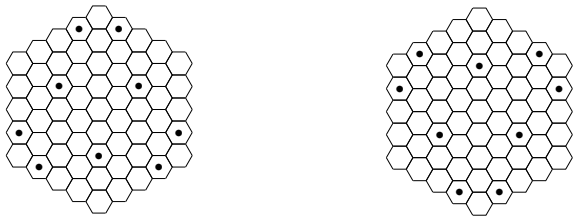


Figure 7.14: The two members of the second equivalence class, where again the second honeycomb array is the vertical reflection of the first.

Table 7.2(a) presents the results by Golomb and Taylor [29] on the enumeration of honeycomb arrays, while Table 7.2(b) exhibits the results of this study, which completes the gaps of the first table up to radius $r = 14$. The new results were derived by using the Costas arrays database [53] as an input to an algorithm implemented in C language (Appendix A). The computer search for honeycomb arrays resulted in four new arrays of radius $r = 7$, that form two equivalence classes, each one consisting of two members.

We continue by presenting the classification of all known honeycomb arrays. In the following, the use of the capital letters A, B and C is arbitrary,

Table 7.2: Enumeration results.

r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...	22
$h(r)$	1	1	0	2	2	?	?	≥ 1	?	?	≥ 1	?	?	≥ 1	?...?	≥ 1
$H(r)$	1	2	0	8	4	?	?	≥ 2	?	?	≥ 2	?	?	≥ 2	?...?	≥ 2

(a) The number of honeycomb arrays of radius r known in 1984.

r	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...	22
$h(r)$	1	1	0	2	2	0	0	3	0	0	1	0	0	1	0	?...?	≥ 1
$H(r)$	1	2	0	8	4	0	0	6	0	0	2	0	0	2	0	?...?	≥ 2

(b) The updated table on the enumeration of honeycomb arrays.

and indicates the different equivalence classes, while the subscript of these letters denotes the radius of the array.

Honeycomb arrays of radius r $= 1$. The only member of the equivalence class of honeycomb arrays of radius $r = 1$ is depicted in Picture 7.15, together with the corresponding Costas array of order $n = 3$, which is generated by the Welsh construction over $\mathbb{F}_5$.

Figure 7.15: The 3×3 Costas array and the corresponding A_{r1} class of the honeycomb arrays.

Honeycomb arrays of radius r $= 3$. The Costas arrays of size $n = 7$ generate 8 honeycomb arrays that are divided into two equivalence classes, A_{r3} and B_{r3}. Both Costas arrays of Figure 7.16 were constructed using the Rickard method with a 3×3 Costas array as a stub. The four honeycomb arrays that were generated by the first Costas array, are $0^0, 120^0, 180^0$ and 240^0 rotation of the hexagon. The remaining rota-

tions appear in the honeycomb array that is produced by the second Costas array of Figure 7.16.

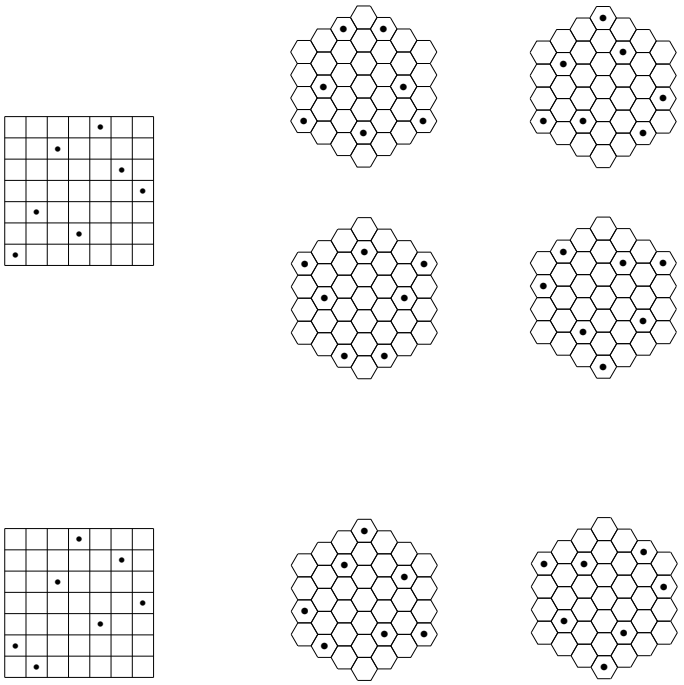


Figure 7.16: The 7×7 Costas arrays and the corresponding A_{r3} class of the honeycomb arrays.

The two remaining honeycomb arrays create the second equivalence class, B_{r3}, which is generated by the Costas array that was constructed using the Lempel-Golomb construction over $\mathbb{F}_{3^2}$ (Figure 7.17).

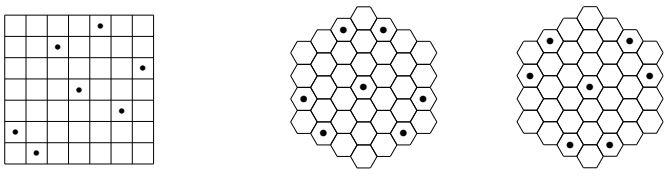


Figure 7.17: The 7×7 Costas array and the corresponding B_{r3} class of the honeycomb arrays.

Honeycomb arrays of radius r $= 4$. Honeycomb arrays of radius $r = 4$ are constructed by Costas arrays of size $n = 9$. Again, we have two equiv-

alence classes, A_{r4} and B_{r4}, each of which consists of two honeycomb arrays. As shown in Figures 7.18 and 7.19, the equivalent honeycomb arrays differ by a rotation of 60^0 degrees. Both equivalence classes are produced by Costas arrays that were constructed using the Lempel method. The first one is over $\mathbb{F}_{11}$ with $\alpha = 6$ as the primitive element, while in the second, the primitive element is $\alpha = 7$.

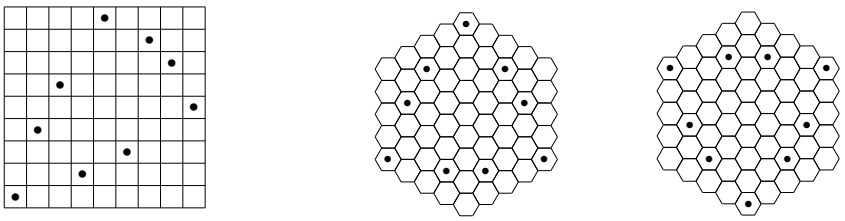


Figure 7.18: The 9×9 Costas array and the corresponding A_{r4} class of the honeycomb arrays.

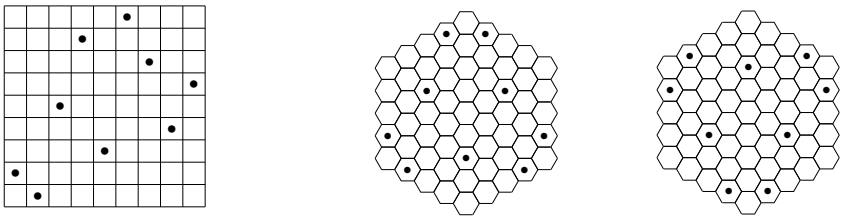


Figure 7.19: The 9×9 Costas array and the corresponding B_{r4} class of the honeycomb arrays.

Honeycomb arrays of radius r $= 7$. In this case, we have in total six honeycomb arrays which are divided into three equivalence classes, A_{r7}, B_{r7} and C_{r7}. Figure 7.20 shows the honeycomb array constructed in 1984 by Golomb and Taylor and Figures 7.21 and 7.22 depict the four new honeycomb arrays that form two equivalences classes, each of which consists of two members. The Costas arrays that generated the new honeycomb arrays were both constructed by the Lempel method over $\mathbb{F}_{17}$ with primitive elements $\alpha = 6$ and $\alpha = 7$, respectively.

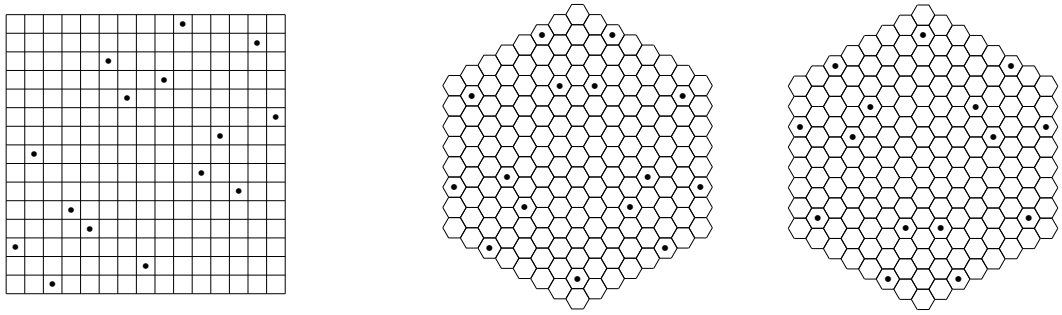


Figure 7.20: The 15×15 Costas array and the corresponding A_{r7} class of the honeycomb arrays.

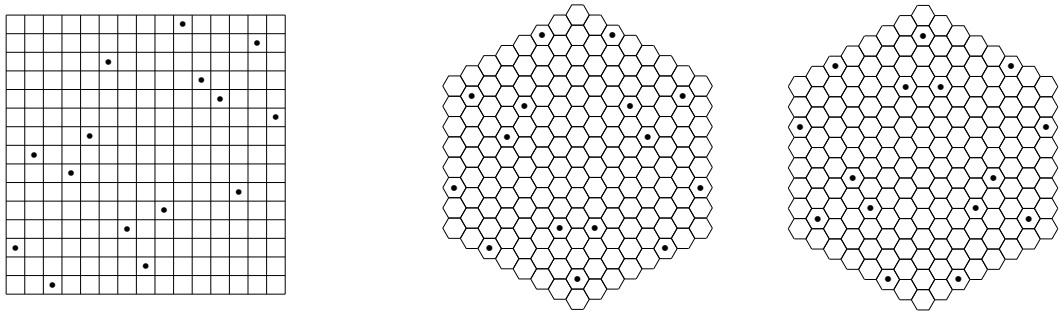


Figure 7.21: The 15×15 Costas array and the corresponding B_{r7} class of the honeycomb arrays.

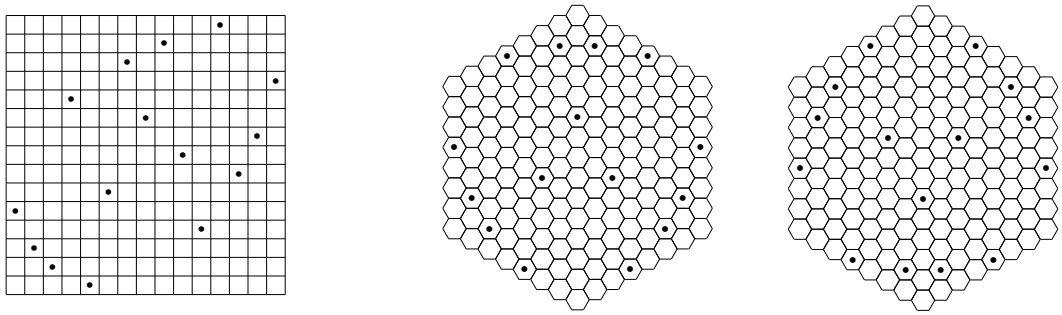


Figure 7.22: The 15×15 Costas array and the corresponding C_{r7} class of the honeycomb arrays.

Honeycomb arrays of radius r $= 10$. There exist only two honeycomb arrays of radius $r = 10$, constructed by the 21×21 Costas array, that was produced by the Lempel construction over $\mathbb{F}_{23}$ with primitive element

$\alpha = 7$. As shown in Figure 7.23 the honeycomb arrays differ by a 60^0 rotation.

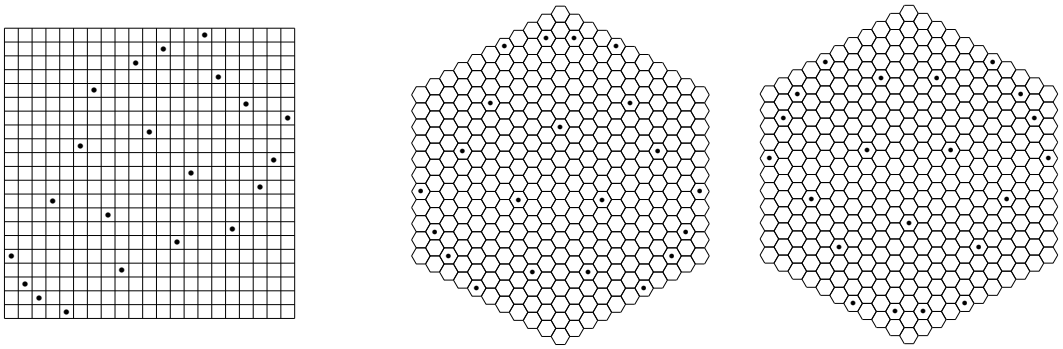


Figure 7.23: The 21×21 Costas array and the corresponding A_{r10} class of the honeycomb arrays.

Honeycomb arrays of radius r $= 13$. As in the previous case, there is only one class of honeycomb arrays of radius $r = 13$, consisting of two members. Figure 7.24 shows the Costas array that generates this class and the two equivalent honeycomb arrays. The 27×27 Costas array was constructed using the Lempel construction over $\mathbb{F}_{29}$ with primitive element $\alpha = 3$.

Honeycomb arrays of radius r $= 22$. The two honeycomb arrays of Figure 7.26 belong to the same equivalence class and were derived from the 45×45 Costas array (Figure 7.25), constructed using the Lempel method, over $\mathbb{F}_{47}$ with primitive element $\alpha = 11$. As it is not known yet whether the list of Costas arrays of order $n = 45$ is complete, there might be more honeycomb arrays of this size.

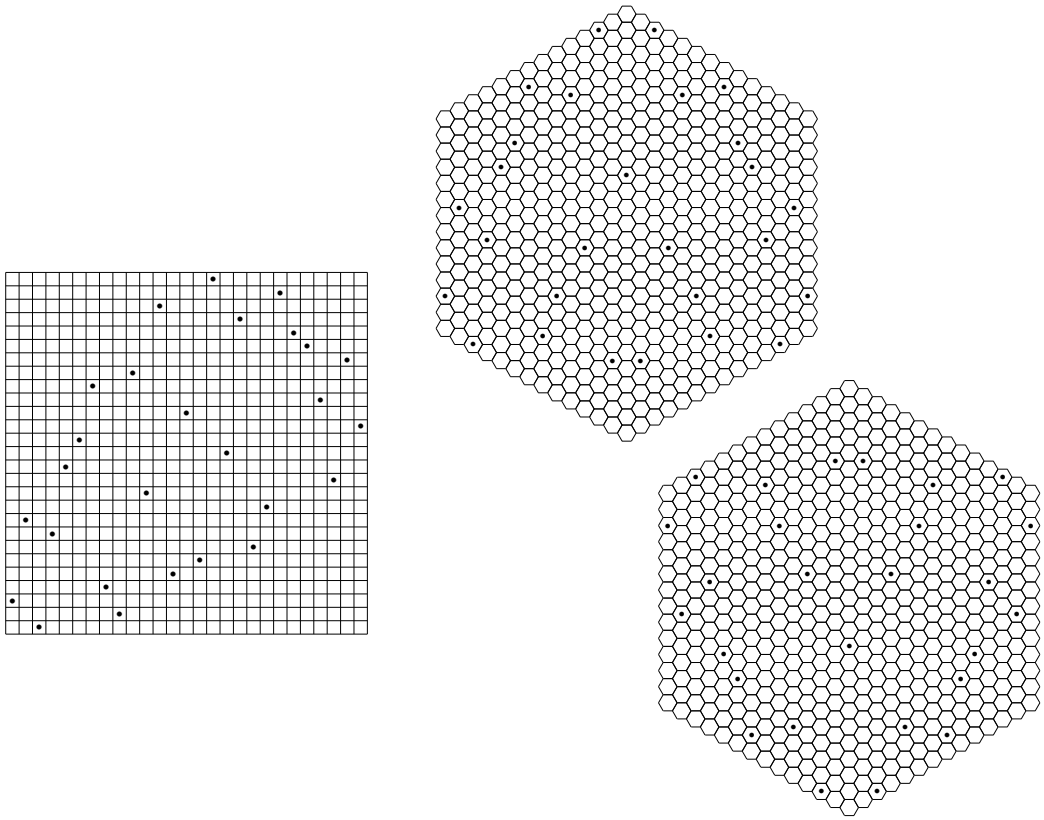


Figure 7.24: The 27×27 Costas array and the corresponding A_{r13} class of the honeycomb arrays.

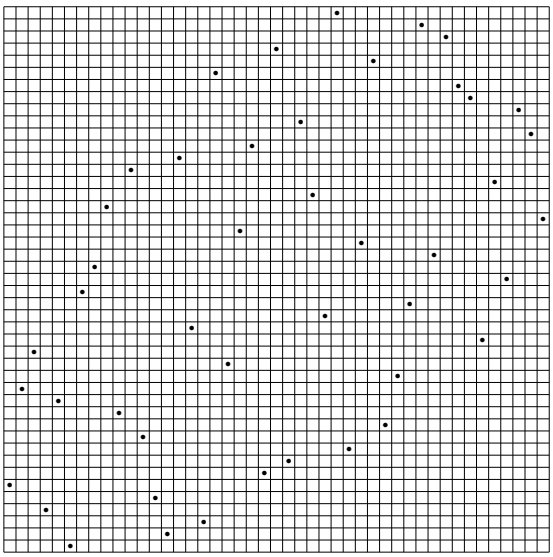
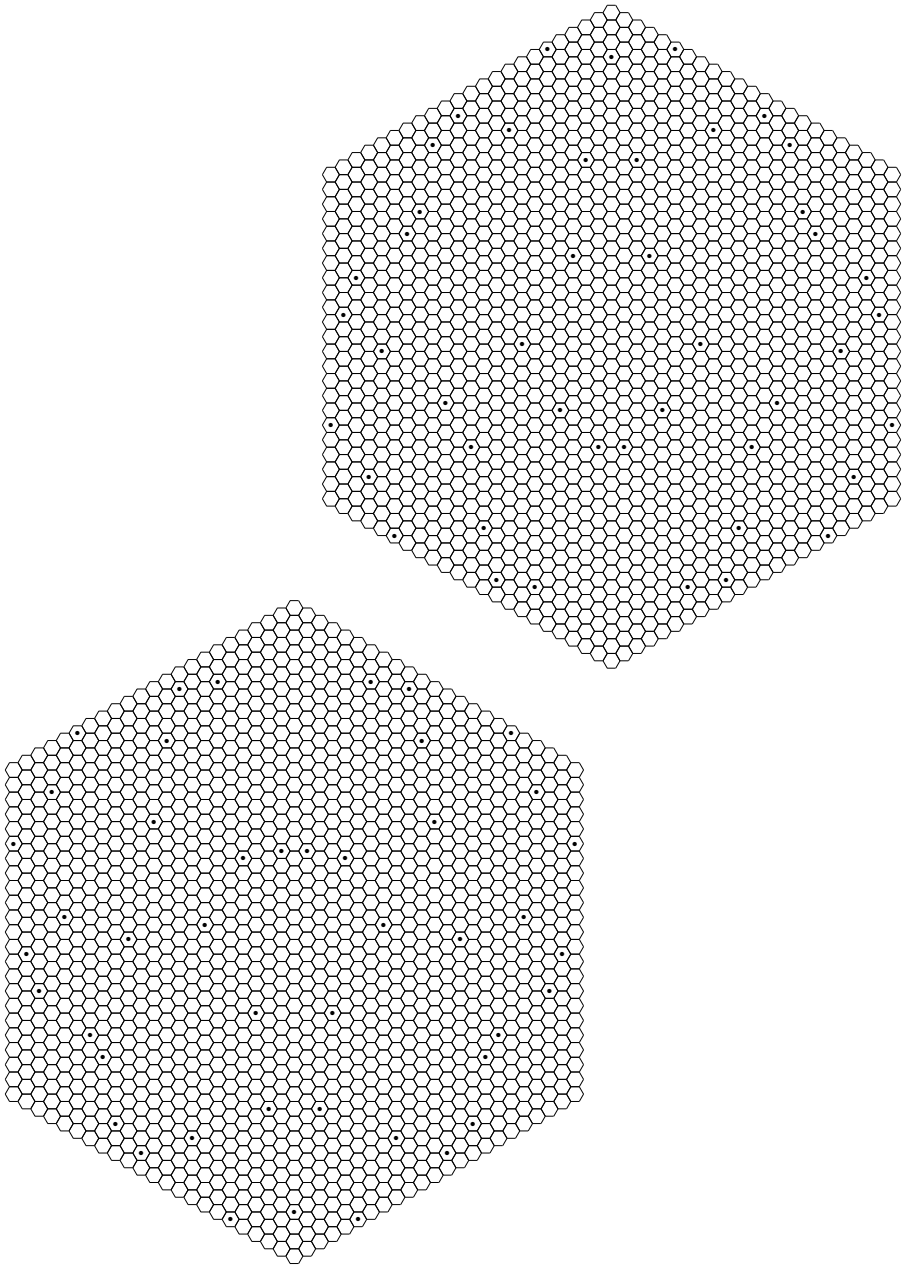


Figure 7.25: The 45×45 Costas array that produces the A_{r22} the honeycomb array.

Figure 7.26: The A_{r22} class of the honeycomb arrays generated from the 45×45 Costas array.



7.4 Concluding Remarks

A more thorough examination of the honeycomb arrays, shows that they inherit the symmetries of the Costas arrays that have generated them. However, all known honeycomb arrays have one symmetry, reflection with respect to the vertical line that passes through the centre, that is not obvious to the corresponding Costas array. As Figure 7.27 shows, this is due to the fact that this symmetry is not a symmetry of the square.

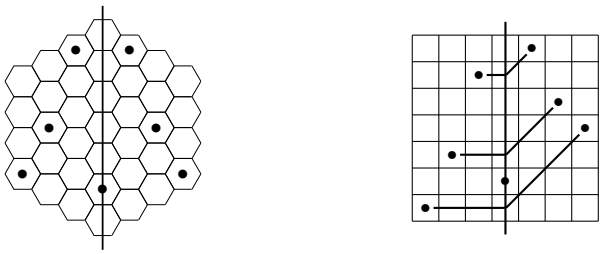


Figure 7.27: The symmetry of the honeycomb arrays to Costas arrays.

Another symmetry that is observed in honeycomb arrays, is the symmetry with respect to the lines defined by the three directions of the hexagonal grid. All known equivalence classes, apart from A_{r3}, consist of honeycomb arrays that posses this symmetry. Figure 7.28 gives an example of this property. Exhaustive search in [13] for honeycomb arrays of radius $r = 31$ and less, having this type of symmetry did not lead to any new arrays.

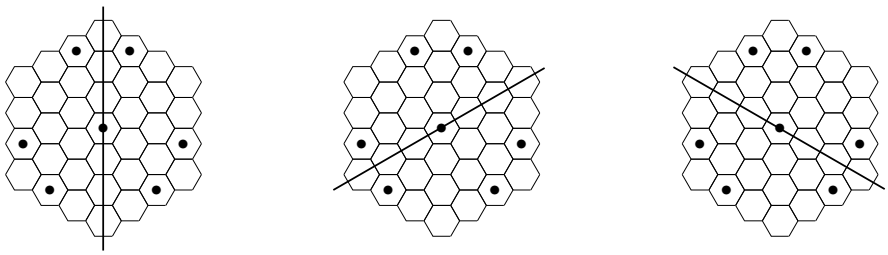


Figure 7.28: The hexagonal symmetries with respect to the lines defined by the three directions of the hexagonal grid.

The entries of Table 7.2(b), and in particular the number $h(r)$, indicate that as the radius increases, the number of honeycomb arrays decreases.

Moreover, the search of honeycomb arrays up to radius $r = 98$ using Costas arrays that have been constructed using algebraic methods (Beard database [7]), was not fruitful. Additional computer searches in [13] for arrays of radius $r \leq 325$, using the Costas arrays listed in Golomb and Taylor paper [29], did not reveal any more examples of honeycomb arrays. All these indications lead to the following conjecture:

Conjecture (Conjecture 1, [13])**.** *The list of known honeycomb arrays is complete. So there are exactly 12 honeycomb arrays, up to symmetry.*

Appendix A

Search of Honeycomb Arrays in C

The following C code determines whether or not a Costas array can lead to honeycomb array. The input to the algorithm is a $n \times n$ Costas array taken from the database that was build by Taylor, Drakakis and Rickard [53]. A necessary condition for the Costas array to generate a honeycomb array, is that n must be an odd number. According to the Golomb-Taylor construction of honeycomb arrays [29], the Costas array must contain non attacking semi-Queens, hence instead of examining the whole $n \times n$ Costas array, the algorithm focuses on the region bounded by the lines $y_1 = i - 1 - r$ and $y_2 = i + n + r$, where $r = (n - 1)/2$. If all n dots of the Costas array lie within this area and in addition, they create a semi-Queen configuration, then the shear-compression method results in a honeycomb array of radius r. These two conditions are examined by the function called **check_and_count**. The function first checks if this area contains all n dots and then examines whether or not the dots follow the non attacking semi-Queens pattern. The first condition, checks if the number of dots in the Costas array that belong to one of the lines between y_1 and y_2, is n. If this is true, then in order to have a honeycomb array, every such line between y_1 and y_2, must contain exactly one dot from the Costas array.

```
/*************************************************************************

                        Honeycomb arrays


 Input : The file c*all.out from the Costas array database.
 Output: The file h*all.txt, which contains the following:

         1. the number of dots in each line y=i-1-r+k, for k=0,...,n-1
         2. (if a honeycomb array was found) the position in the Costas
            array file, of the Costas array that led to the honeycomb
            array.

  total : The number of Costas arrays.
  n     : The size of Costas array (the number of dots)
  y[]   : The array that contains the dot positions on the Costas array

*************************************************************************/



#include<stdio.h>
#include<math.h>
#include<time.h>

#define nmax 30
#define total 2

void ini(int);
int check_and_count(int,int);

FILE *fr,*fw;

int E[nmax],y[nmax];


 void main()
{
 int i,j,r,n,dots;
 double cpu_time;
 clock_t start, end;

 start=clock();
```

```
n=nmax-1;

fr=fopen("c29.out","r+");

if(n%2==1)  fw=fopen("h7all.txt","w+");

fprintf(fw,"\t  Number of dots in diagonals \t\t  Results    Row no.\n\n");


for(j=1; j<=total; j++)
{
  for(i=1; i<=n; i++)
  {
    fscanf(fr,"%d",&y[i]);
  }

  fprintf(fw,"  ");

  if(n%2==1)
  {
    r=(n-1)/2;
    ini(n);
    dots=check_and_count(n,r);

    if(dots==n)
    {
      printf(" Honeycomb Array in row j=%d \n",j);
      fprintf(fw,"\t  Honeycomb   %d \n",j);
    }
    else
    {
      fprintf(fw,"\t      -         %d \n",j);
    }
   }
}

fclose(fr);
fclose(fw);

end=clock();
cpu_time=((double)(end-start))/CLOCKS_PER_SEC;
printf(" cpu time = %f\n", cpu_time);

}
```

```
/*************************************************************************
                              FUNCTIONS
*************************************************************************/


void ini(int n)
{
  int i;

  for(i=1; i<=n; i++)
  {
    E[i]=0;
  }
}


int check_and_count(int n, int r)
{
  int i,k,flag,num;

  for(i=1; i<=n; i++)
  {
    if(i-1-r < y[i] < i+n-r)
    {
      flag=0;
      k=0;
      do
      {
        if(y[i]==i-1-r+k)              // checks whether there is a dot in y[i]
        {
          E[k]=E[k]+1;
          flag=1;
        }
        k=k+1;
      }
      while(k<=n && flag==0);
    }
  }

  num=0;

  for(i=1; i<=n; i++)
```

```
  {
    if(E[i]==1)                    // checks if in line E[i] occurs only one dot
    {
      num=num+1;                   // counts the total number of dots
    }
    fprintf(fw, " %d ", E[i]);
  }

  return(num);
}
```

Bibliography

[1] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[2] N. Alon, G. Cohen, M. Krivelevich, and S. Litsyn. Generalized hashing and applications to digital fingerprinting. In *Proceedings of the 2002 IEEE International Symposium on Information Theory*, page 436, 2002.

[3] N. Alon and U. Stav. New bounds on parent-identifying codes: The case of multiple parents. *Combinatorics, Probability & Computing*, 13(6):795–807, 2004.

[4] A. Barg, G. R. Blakley, and G. A. Kabatiansky. Digital fingerprinting codes: problem statements, constructions, identification of traitors. *IEEE Transactions on Information Theory*, 49(4):852–865, 2003.

[5] A. Barg, G. Cohen, S. Encheva, G. A. Kabatiansky, and G. Zémor. A hypergraph approach to the identifying parent property: The case of multiple parents. *SIAM Journal on Discrete Mathematics*, 14(3):423–431, 2001.

[6] A. Barg and G. A. Kabatiansky. A class of I.P.P. codes with efficient identification. *Journal of Complexity*, 20(2-3):137–147, 2004.

[7] J. K. Beard. Costas arrays database. http : //jameskbeard.com/jameskbeard/ (last accessed October 2011).

[8] S. R. Blackburn. Combinatorial schemes for protecting digital content. In *Surveys in Combinatorics 2003*, volume 307, pages 43–78. Cambridge University Press, 2003.

[9] S. R. Blackburn. Frameproof codes. *SIAM Journal on Discrete Mathematics*, 16(3):499–510, 2003.

[10] S. R. Blackburn. An upper bound on the size of a code with the k-identifiable parent property. *Journal of Combinatorial Theory Series A*, 102:179–185, 2003.

[11] S. R. Blackburn, T. Etzion, K. M. Martin, and M. B. Paterson. Two-dimensional patterns with distinct differences: constructions, bounds, and maximal anticodes. *IEEE Transactions on Information Theory*, 56(3):1216–1229, 2010.

[12] S. R. Blackburn, T. Etzion, and S. Ng. Traceability codes. *Journal of Combinatorial Theory Series A*, 117(8):1049–1057, 2010.

[13] S. R. Blackburn, A. Panoui, M. B. Paterson, and D. R. Stinson. Honeycomb arrays. *Electronic Journal of Combinatorics*, 17(1):R172, 2010.

[14] D. Boneh and M. K. Franklin. An efficient public key traitor tracing scheme. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 338–353. Springer-Verlag, 1999.

[15] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data (extended abstract). In *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '95, pages 452–465. Springer-Verlag, 1995.

[16] H. Chabanne, D.H. Phan, and D. Pointcheval. Public traceability in traitor tracing schemes. In *Proceedings of the 24th Annual International*

Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT '05, pages 542–558. Springer-Verlag, 2005.

[17] Y. M. Chee. *Turán-type problems in group testing, coding theory, and cryptography*. PhD thesis, Department of Computer Science, University of Waterloo, Canada, 1996.

[18] B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '94, pages 257–270. Springer-Verlag, 1994.

[19] J. P. Costas. Project Medior – A medium-oriented approach to sonar signal processing. Technical report, Lockheed Martin Marine Systems and Sensors, Syracuse, NY, USA, 1966.

[20] N. G. de Bruijn, C. Tengbergen, and D. Kruyswijk. On the set of divisors of a number. *Nieuw Archief Wiskunde*, 23(2):191–193, 1951.

[21] K. Drakakis. A review of Costas arrays. *Journal of Applied Mathematics*, 2006:1–32, 2006.

[22] K. Drakakis, F. Iorio, and S. Rickard. The enumeration of Costas arrays of order 28. *Information Theory Workshop (ITW), 2010 IEEE*, pages 1–5, 2010.

[23] K. Drakakis, F. Iorio, S. Rickard, and J. Walsh. Results of the enumeration of Costas arrays of order 29. *Advances in Mathematics of Communications*, 5(3):547–553, 2011.

[24] K. Drakakis, S. Rickard, J. K. Beard, R. Caballero, F. Iorio, G. O'Brien, and J. Walsh. Results of the enumeration of Costas arrays of order 27. *IEEE Transactions on Information Theory*, 54(10):4684–4687, 2008.

[25] S. Encheva and G. Cohen. Frameproof codes against limited coalitions of pirates. *Theoretical Computer Science*, 273(1-2):295–304, 2002.

[26] K. Engel. *Sperner Theory (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 1997.

[27] P. Erdős, C. Ko, and R. Rado. Intersection theorems for systems of finite sets. *The Quarterly Journal of Mathematics*, 12(1):313–320, 1961.

[28] P. Frankl and R. L. Graham. Old and new proofs of the Erdős-Ko-Rado thoerem. *Journal of Sichuan University Natural Science Edition*, 26:112–122, 1991.

[29] S. W. Golomb and H. Taylor. Constructions and properties of Costas arrays. *Proceedings of the IEEE*, 72(9):1143–1163, 1984.

[30] H. D. L. Hollmann, J. H. van Lint, J. P. Linnartz, and L. M. G. M. Tolhuizen. On codes with the identifiable perent property. *Journal of Combinatorial Theory Series A*, 82(2):121–133, 1998.

[31] G. O. H. Katona. Intersection theorems for systems of finite sets. *Acta Mathematica Hungarica*, 15:329–337, 1964.

[32] G. O. H. Katona. Two applications (for search theory and truth functions) of Sperner type theorems. *Periodica Mathematica Hungarica*, 3:19–26, 1973.

[33] A. Kiayias and M. Yung. Traitor tracing with constant transmission rate. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 450–465. Springer-Verlag, 2002.

[34] K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, EUROCRYPT '98, pages 145–157. Springer-Verlag, 1998.

[35] K. Kurosawa and T. Yoshida. Linear code implies public-key traitor tracing. In *Proceedings of the 5th International Workshop on Practice and*

Theory in Public Key Cryptosystems, PKC '02, pages 172–187. Springer-Verlag, 2002.

[36] L. Liu and H. Shen. Explicit constructions of separating hash families from algebraic curves over finite fields. *Designs, Codes and Cryptography*, 41(2):221–233, 2006.

[37] E. C. Milner. A combinatorial theorem on systems of sets. *Journal of The London Mathematical Society*, 43:204–206, 1968.

[38] D. Naccache, A. Shamir, and J. P. Stern. How to copyright a function? In *Proceedings of the Second International Workshop on Practice and Theory in Public Key Cryptography*, PKC '99, pages 188–196. Springer-Verlag, 1999.

[39] G. Nivasch and E. Lev. Non attacking Queens on a triangle. *Mathematics Magazine*, 78:399–403, 2005.

[40] B. Pfitzmann. Trials of traced traitors. In *Proceedings of the First International Workshop on Information Hiding*, pages 49–64. Springer-Verlag, 1996.

[41] B. Pfitzmann and M. Schunter. Asymmetric fingerprinting. In *Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '96, pages 84–95. Springer-Verlag, 1996.

[42] B. Pfitzmann and M. Waidner. Asymmetric fingerprinting for larger collusions. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 151–160. ACM, 1997.

[43] S. Rickard. Searching for Costas arrays using periodicity properties. In *IMA International Conference on Mathematics in Signal Processing*, The Royal Agricultural College, Cirencester, 2004.

[44] J. Schönheim. On a problem of Purdy related to Sperner systems. *Canadian Mathematical Bulletin*, 17:135–136, 1974.

[45] A. Silverberg, J. Staddon, and J. L. Walker. Applications of list decoding to tracing traitors. *IEEE Transactions on Information Theory*, 49(5):1312–1318, 2003.

[46] E. Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928.

[47] J. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. *IEEE Transactions on Information Theory*, 47(3):1042–1049, 2001.

[48] D. R. Stinson and P. Sarkar. Frameproof and IPP codes. In *Proceedings of the Second International Conference on Cryptology in India: Progress in Cryptology*, INDOCRYPT '01, pages 117–126. Springer-Verlag, 2001.

[49] D. R. Stinson, T. van Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, 86(2):595–617, 2000.

[50] D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM Journal on Discrete Mathematics*, 11(1):41–53, 1998.

[51] D. R. Stinson and G. M. Zaverucha. Some improved bounds for secure frameproof codes and related separating hash families. *IEEE Transactions on Information Theory*, 54(6):2508–2514, 2008.

[52] H. Taylor. Non-attacking Rooks with distinct differences. Technical Report CSI-84-03-2, EE Systems, University of Southern California, 1984.

[53] K. Taylor, K. Drakakis, and S. Rickard. Costas arrays database. http : //www.costasarrays.org/ (last accessed October 2011).

[54] V. D. Tô, R. Safavi-Naini, and Y. Wang. A 2-secure code with efficient tracing algorithm. In *Proceedings of the Third International Conference on Cryptology: Progress in Cryptology*, INDOCRYPT '02, pages 149–162. Springer-Verlag, 2002.

[55] D. Tonien and R. Safavi-Naini. Explicit construction of secure frame-proof codes. *International Journal of Pure and Applied Mathematics*, 6(3):343–360, 2003.

[56] P. Vaderlind, R. Guy, and L. C. Larson. *The inquisitive problem solver*. Mathematical Association of America, 2002.

[57] C. Xing. Asymptotic bounds on frameproof codes. *IEEE Transactions on Information Theory*, 48(11):2991–2995, 2002.