On secret sharing schemes and linear codes

Romar Basillaje Dela Cruz

2013

Romar Basillaje Dela Cruz. (2013). On secret sharing schemes and linear codes. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54695>

<https://doi.org/10.32657/10356/54695>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ON SECRET SHARING SCHEMES
AND LINEAR CODES**

**ROMAR BASILLAJE DELA CRUZ
SCHOOL OF PHYSICAL AND MATHEMATICAL
SCIENCES**

2013

ON SECRET SHARING SCHEMES AND LINEAR CODES

ROMAR BASILLAJE DELA CRUZ

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2013

Acknowledgements

I would like to express my tremendous gratitude to my supervisors Assoc. Prof. Wang Huaxiong, Prof. Ling San and Dr. Patrick Solé for their infinite patience and wisdom, for their encouragement and for their generosity. I'm extremely grateful to my collaborators Dr. Annika Meyer, Dr. Gao Ying, Prof. R.E.L. Aldred, Prof. Carsten Thomassen and Dr. Adel Alahmadi for sharing their knowledge and expertise. I'm also indebted to Nanyang Technological University for the NTU Research Scholarship, to the French Embassy in Singapore for the Merlion PhD Grant, and to the University of the Philippines Diliman for the UP Modernization - Doctoral Studies Grant. I gratefully acknowledge the assistance provided by the following institutions: Division of Mathematical Sciences and SPMS of NTU, Laboratoire I3S of CNRS-UNSA, Dept Comelec of Telecom-Paristech and the Institute of Mathematics of UP Diliman. Special thanks to all my friends and colleagues for all the help and the adventures. Most importantly, I dedicate this work to my beloved wife Jen, our son Rafael and to our families.

Abstract

A secret sharing scheme is a method of distributing a secret among a group of participants in such a way that only certain subgroups are authorized to know the secret. Secret sharing schemes have been used in many areas of cryptography such as secure information storage, threshold cryptographic protocols, secure message transmission, secure multiparty computation, and generalized oblivious transfer. This thesis deals with secret sharing schemes constructed using linear codes.

First, we consider the secret sharing schemes based on linear codes constructed using the approach by Karnin, Greene and Hellman, by Bertilsson and Ingemarsson and by Massey. In this approach, the minimal authorized groups correspond to certain minimal codewords of the dual of the underlying code. We study three topics related to secret sharing schemes based on linear codes.

The first topic is on the access structure and multiplicativity of linear secret sharing schemes based on codes from complete graphs. Multiplicative linear secret sharing schemes are used in the construction of secure multiparty computation protocols. The characterization of access structures of ideal multiplicative schemes has not been solved completely. We describe the access structure of the schemes based on cut-set and cycle codes and show that these schemes cannot be multiplicative. We then show

that the class of access structures based on odd cycles cannot be realized by ideal multiplicative linear secret sharing schemes over any finite field. This can be seen as a contribution to the characterization of access structures of ideal multiplicative schemes. The access structure based on odd cycles corresponds to the scheme based on the dual of the extended cycle code. Finally, we show that we can obtain ideal multiplicative linear secret sharing scheme based on the dual of an augmented extended cycle code.

The second topic is a generalization of secret sharing schemes based on linear codes. In this generalization, the secret consists of multiple field elements. We give a characterization of the groups in the access structure using the dual code. We then use a generalized joint weight enumerator to describe the access structure.

The third topic is on the maximum number of minimal codewords in binary linear codes. This subject has been studied before but in the setting of cycles in graphs and of circuits in matroids. We prove several upper and lower bounds on the maximum number of minimal codewords in binary linear codes given the length and dimension. We make the bounds explicit for code of small length and dimension. Exact values are given for small cycle codes of graphs. Some asymptotic results are presented including the asymptotic behavior for cycle codes of graphs.

We also consider another type of secret sharing schemes constructed using linear codes. These are the so-called cheating-immune secret sharing schemes. A secret sharing scheme that is cheating-immune prevents a cheater, who submits a corrupted share, from gaining an advantage in knowing the secret over the honest participants. There are only a few known constructions of cheating-immune schemes and all of these are for the (n, n) -access structure. We revisit two methods, that uses linear

codes, to construct Boolean functions satisfying multiple cryptographic criteria. We show that these methods can be used to build new cheating-immune (n, n) -secret sharing schemes. We also revisit two general constructions of secret sharing schemes using cumulative arrays and apply them to build cheating-immune (t, n) -threshold secret sharing schemes.

Contents

Acknowledgements	iii
Abstract	v
1 Introduction	1
1.1 Secret sharing schemes and linear codes	1
1.2 Our contributions	6
1.2.1 Secret sharing schemes based on graphical codes	6
1.2.2 General secret sharing scheme based on linear codes	7
1.2.3 Minimal codewords in binary linear codes	8
1.2.4 Cheating-immune secret sharing schemes	9
1.3 Organization of this thesis	12
2 Preliminaries	13
2.1 Linear codes	13
2.2 Graphs	25
2.3 Matroids	27
2.4 Entropy	31

2.5	Summary	33
3	Secret Sharing Schemes	35
3.1	Definition	35
3.2	Some constructions	40
3.2.1	Shamir threshold scheme	40
3.2.2	KGH (n, n) -threshold scheme	41
3.2.3	Secret sharing schemes from cumulative arrays	42
3.3	Secret sharing schemes and matroids	43
3.4	Linear secret sharing schemes and monotone span programs	45
3.5	Secret sharing schemes from linear codes	50
3.5.1	Scheme I	50
3.5.2	Scheme II	51
3.6	Multiplicative linear secret sharing schemes	56
3.7	Non-perfect secret sharing schemes	61
3.8	Summary	63
4	Secret sharing schemes based on graphical codes	65
4.1	Secret sharing schemes based on cut-set and cycle codes	66
4.2	Access structure based on odd cycles	69
4.2.1	Definition and some properties	69
4.2.2	Appropriate matroid	70
4.2.3	LSSS realizing the access structure based on odd cycles	72
4.2.4	Multiplicativity	74
4.3	Ideal multiplicative LSSS based on graphical codes	76

4.3.1	Access structure	76
4.3.2	Linear secret sharing scheme realizing Γ_{ocy}	78
4.3.3	Multiplicativity	80
4.4	Conclusion	83
5	A generalization of SSS based on linear codes	85
5.1	General SSS based on linear codes	86
5.2	Access Structure	87
5.3	Relation between access structure and joint weight enumerator	90
5.4	Binary self-dual codes	94
5.5	Invariant theory	98
5.6	Conclusion	100
6	The maximum number of minimal codewords in binary codes	101
6.1	Bounds	103
6.1.1	Upper bounds	103
6.1.2	Lower bounds	106
6.2	Tabulating $W(n, k)$	107
6.2.1	Monotonicity properties	107
6.2.2	Exact values	108
6.2.3	Intersecting codes	109
6.2.4	Table of $W(n, k)$	110
6.2.5	Cycle codes of graphs	110
6.3	Asymptotic analysis	112
6.4	Conclusion	118

7	Cheating-immune SSS from codes and cumulative arrays	121
7.1	Secret sharing schemes with cheaters	122
7.1.1	Tompa-Woll attack	122
7.1.2	Security of SSS from linear codes against cheating	124
7.1.3	Other SSS secure against cheating	124
7.2	Cheating-immune secret sharing schemes	125
7.3	Cheating-immune schemes and Boolean functions	129
7.4	Cheating-immune schemes from binary linear codes	130
7.4.1	Construction of cheating-immune schemes	131
7.4.2	Existence of cheating-immune schemes	133
7.5	Strictly cheating-immune schemes from binary linear codes	134
7.5.1	Construction of strictly cheating-immune schemes	135
7.5.2	Existence of strictly cheating-immune schemes	138
7.6	Cheating-immune threshold schemes	138
7.6.1	Cheating-immune threshold schemes from cumulative arrays .	139
7.6.2	Cheating-immune threshold schemes from generalized cumula- tive arrays	141
7.7	Conclusion	146
8	Concluding Remarks	149
	Bibliography	153
	List of Works	169
	List of Figures	171

Chapter 1

Introduction

1.1 Secret sharing schemes and linear codes

A *secret sharing scheme (SSS)* is a process of distributing access to a secret among a set of participants in such a way that only certain subsets of participants can reconstruct the secret. This process begins with a *dealer* who knows the secret and who generates the *shares*. Each participant will then receive a share from the dealer. Afterwards, a subset of participants that is authorized to know the secret will jointly use their shares to reconstruct it.

As a motivation, consider a company that uses digital signatures (this application was suggested by Shamir in [100]). The president is the only one who knows the secret key but in case of his/her absence, the company should still be able to digitally sign documents through the vice-presidents. Now the president does not want to give a copy of the key to any of the vice-presidents because of trust issues. A solution is to split the key in such a way that the reconstruction of the secret key requires the

collaboration of at least two vice-presidents.

A solution is to use the Shamir (t, n) -threshold secret sharing scheme. Take a random polynomial of degree one (or a line) such that the constant term is equal to the secret. Each vice-president will be given a point in the line. We can see that any set of two points is enough to determine the secret. If one of the vice-presidents tries to determine the secret then he/she can only guess the value of the secret (since for any point p there is a unique line passing through p and the vice-president's point).

Secret sharing schemes have been an integral part of many cryptographic systems. Some of these systems are secure information storage, threshold cryptographic protocols, secure message transmission, secure multiparty computation, and generalized oblivious transfer.

The example above shows an application of secret sharing schemes in secure information storage and also in threshold cryptography. Another example for threshold cryptography can be found in [42]. This involves a public key cryptosystem. An organization has a single public key for encryption but the private key is shared among its members using a threshold secret sharing scheme. In this way, an encrypted message can only be read by a group with sufficiently many members.

In a secure message transmission scheme [43, 45], we have a sender who wants to transmit a message to a receiver by sending information over a set of channels. There is also an adversary who can corrupt the information sent through some of the channels. The goals of the adversary are to learn the message sent and to prevent the receiver from knowing the correct message. To send a message securely, the sender splits the message using a secret sharing scheme with an error-correcting capability. The shares generated are distributed among the channels. The error-correcting capa-

bility enables the sender and the receiver to identify the channels under the control of the adversary and to correct any modifications introduced by the adversary.

The main idea of secure multiparty computation [14, 37, 38] is to enable a set of parties to jointly compute a function of their inputs in such a way that the inputs remain secret (except to the party who owns it). Secret sharing schemes with multiplicative properties are used to construct protocols for secure multiplication. More details can be found in Chapter 3.

A protocol for generalized oblivious transfer involves two parties: a sender and a receiver. The sender has a set of messages M and there is a collection N of subsets of M . The receiver can get any subset of messages in N . The security requirements are as follows: the sender must not obtain any information on the subset of messages that the receiver selected while the receiver must not learn any information on the messages that are not in the selected subset. Protocols that use secret sharing schemes can be found in [101, 107].

Secret sharing schemes were first proposed independently by Shamir [100] and Blakley [19] in 1979. By using polynomials over finite fields (Shamir) and finite geometries (Blakley), the two authors constructed schemes for the threshold access structure. Ito, Saito and Nishizeki [61] introduced general access structures and demonstrated that any monotone access structure can be realized by a perfect secret sharing scheme. The main drawback of this general construction is that the size of shares is generally very large. It is an open problem to construct efficient schemes realizing general access structures. Other important constructions are the vector space construction by Brickell [27] and the monotone span program construction by Karchmer and Wigderson [66].

Brickell and Davenport [28] showed the connection between ideal perfect secret sharing schemes and matroids. The characterization of ideal schemes is one of the main open problems in the theory of secret sharing schemes. It is connected to the problem of representability of matroids over finite fields.

The relation between secret sharing schemes and linear codes was first noted by McEliece and Sarwate. In [85], they presented a construction of threshold schemes using Reed-Solomon codes. The Shamir threshold scheme is a special case of that construction. This construction can be applied to any linear code. They also pointed out that the error-correcting capability of linear codes can be used to deal with the case when some of the shares are tampered. This construction is also equivalent to Brickell's vector space construction and to the monotone span program construction by Karchmer and Wigderson.

A different kind of approach in constructing secret sharing schemes from linear codes was studied by Karnin, Greene and Hellman [67], by Bertilsson and Ingermarsson [16], and by Massey [83, 84]. This approach was used in [67] to construct threshold schemes from MDS codes while more general schemes were considered in [16] and [83, 84]. Chen and Cramer [32] applied this method to construct schemes using algebraic-geometric codes. The resulting secret sharing schemes have special properties that are useful in secure multiparty computation.

Massey showed in [83] that the access structure of the scheme based on a linear code can be determined by the dual code. He proved that there is a 1-1 correspondence between the minimal authorized sets and certain minimal codewords of the dual code. This result sparked interest on the problem of determining the minimal codewords of a linear code.

Since the publication of the pioneering papers of Shamir and Blakley, a lot of research has been done on modifications and extensions of the concept of secret sharing schemes.

Secret sharing schemes without the presence of the dealer were considered in [64]. In these schemes, the participants execute a protocol to generate the shares. Traditionally, it is assumed that the dealer and participants communicate through secure private channels. The reconstruction of the secret using only a public channel was studied in [12]. Secret sharing on partial broadcast channels was investigated in [98, 111].

Many authors work on secret sharing schemes with additional processes or capabilities. One example is called proactive secret sharing [57] where shares are updated periodically. If the shares are not updated, it is possible for an adversary to obtain sufficiently many shares during the lifetime of the scheme. Another example is the updating of parameters of a threshold scheme after the initial setup [8, 18, 79, 81]. The parameters to be updated include the number of participants and the threshold number. The authors in [22] studied a secret sharing scheme where a particular access structure and secret can be activated from a set of access structures and secrets. There is also the class of secret sharing schemes that can deal with dishonest entities (participants or dealer). Some of these schemes can be found in [34, 69, 85, 89, 93, 95]. To deal with dishonest entities, these schemes detect cheating, correct the errors introduced by cheating, or prevent the cheaters from gaining an advantage in the recovery of the secret. In Chapter 7, we are going to consider one such scheme, called cheating-immune secret sharing schemes.

1.2 Our contributions

The results of this thesis are as follows.

1.2.1 Secret sharing schemes based on graphical codes

Cramer, Damgård and Maurer demonstrated in [36] that linear secret sharing schemes (LSSS) can be a fundamental primitive in secure multiparty computation protocols (MPC). Informally speaking, a secure MPC protocol [54, 115] involves n players who jointly compute a function of their secret inputs. The protocol must guarantee the correctness of the output and the privacy of the inputs even when some of the players are corrupted. Secure computation of a function can be reduced to secure addition and secure multiplication. Secure addition can be performed using the linearity of the scheme. To do secure multiplication, the authors in [36] introduced the concept of multiplicative LSSS.

A multiplicative LSSS can be constructed from any LSSS for the same access structure, though in general, the process involves doubling the share size [36]. It is an open problem to determine the cases in which the increase in the share size is necessary and the cases in which it is not. A special case of this problem is the characterization of access structures that admit ideal multiplicative LSSS. This problem was considered in [37, 91] for some self-dual access structures. Many authors also considered the construction of ideal multiplicative LSSS [32, 37, 68, 73, 87]. Some examples are the Shamir threshold scheme, algebraic geometric secret sharing schemes and LSSS based on self-dual codes. The authors in [73] constructed ideal multiplicative LSSS based on graph connectivity.

We study the access structure and multiplicativity of linear secret sharing schemes based on codes from complete graphs. First, we describe the access structure of the schemes based on cut-set and cycle codes and show that these schemes cannot have the multiplicative property. Second, we show that the class of access structures based on odd cycles cannot be realized by ideal multiplicative linear secret sharing schemes over any finite field. This can be seen as a contribution to the characterization of access structures of ideal multiplicative schemes. The access structure based on odd cycles corresponds to the scheme based on the dual of the extended cycle code. Finally, we show that we can obtain ideal multiplicative linear secret sharing scheme based on the dual of an augmented extended cycle code.

1.2.2 General secret sharing scheme based on linear codes

The result of Massey [83] showing the 1-1 correspondence between minimal authorized sets and minimal codewords leads to the following problem: Given a secret sharing scheme based on a linear code, determine its access structure or the set of minimal codewords. In general, this is a difficult problem in coding theory.

Some authors described the access structure by obtaining the weight distribution of the codewords of the dual code with first coordinate equal to 1. These codewords correspond to authorized sets but are not necessarily minimal. The description contains the number of authorized sets of different sizes. This is how Dougherty, Mesnager and Solé [48] described the access structure of secret sharing schemes based on self-dual codes. They used the properties of designs arising from the codes, the Jacobi weight enumerator and invariant theory.

In this thesis, we study the access structure of a generalization of secret sharing schemes based on linear codes. In this generalized secret sharing scheme, the secret consists of multiple field elements. This scheme is a special case of the one studied by Blakley and Kabatianskii. In [20], they gave conditions on when a given subset of participants is authorized, unauthorized or can have partial information about the secret. In [33], Chen *et al.* gave a lower bound on the size of authorized sets and an upper bound on the size of unauthorized sets in terms of the distances of the codes and its dual.

First, we characterize the authorized subsets of participants in terms of the codewords of the dual code. This extends the result by Massey on the classical case. Then we analyze the access structure of the schemes based on binary self-dual codes by extending some of the techniques in [48]. In particular, we use a generalized joint weight enumerator to obtain information on the number of authorized sets of different sizes in the access structure.

1.2.3 Minimal codewords in binary linear codes

It is important to determine the minimal codewords of a linear codes mainly because of its applications to secret sharing schemes and to decoding algorithms of linear codes. However, to determine the minimal codewords is generally a very difficult problem and it is solved only for a few classes of codes (for instance, see [1, 5, 24, 44, 72, 116]). It is harder than the problem of knowing the weight distribution of linear codes which was shown to be NP-complete [15].

Given the length n and dimension k , we study the maximum number of minimal

codewords of binary linear codes. If we restrict to cycle codes, this problem was considered before by Entringer and Slater in [49] where they asked for the maximum number of cycles in connected graphs. A general version of this problem was studied by Dósa, Szalkai and Laflamme in [46] where they investigated the maximum number of circuits of matroids of a given size and rank.

In this thesis, we present some upper and lower bounds on the maximum number of minimal codewords of binary linear codes of a given length and dimension. We compute explicit bounds for $n \leq 15$ and $1 \leq k \leq 15$. We give exact values for cycle codes of graphs for $3 \leq n \leq 15$ and $1 \leq k \leq 13$. We also discuss the asymptotic behavior of the quantity in question and determine the exact behavior for cycle codes.

1.2.4 Cheating-immune secret sharing schemes

Traditionally, it is assumed that the entities in a secret sharing scheme do not deviate from the protocol. There are many scenarios though, where this assumption is not realistic. We look into a secret sharing model where some of the participants are dishonest. We assume that when an authorized group wants to know the secret, the members submit their shares to a combiner. The combiner computes the secret and then transmits it to the members of the group.

We consider the setting where the dealer and the combiner are honest but participants can cheat by submitting corrupted shares during reconstruction. In this model, Tompa and Woll [108] showed that in the Shamir threshold scheme, when a participant submits an invalid share then the combiner will return an incorrect secret. As a result, the honest participants are prevented from knowing the correct secret.

More importantly, the cheater can use the incorrect secret to determine the correct one. This attack also applies to any linear secret sharing scheme.

There are many types of secret sharing schemes that deal with this kind of attack. These schemes offer different forms of security but the added security comes with a price. In general, the tradeoff can be classified into two types.

The first type is the need for extra shares to reconstruct a secret. Recall that the shares of a minimum authorized set is enough to recover the secret. The presence of a certain number of extra shares makes it possible to (at least) detect cheating. For example, the code-based secret sharing schemes use the error-detecting/correcting property of the underlying linear code to deal with cheating [67, 83, 85, 96]. These schemes can detect the presence of invalid shares or even correct the invalid shares and recover the secret.

The second type is the generation of shares with extra information which leads to large shares. An example is a verifiable secret sharing scheme (VSSS) introduced in [34, 95]. The extra information in the shares in a VSSS are used to verify the validity of the shares. VSSS also deal with the case when the dealer is also dishonest. Other examples can be found in [69, 89, 97, 108]. These schemes also require shares with extra information to be used for cheater detection or identification.

In this thesis, we study cheating-immune secret sharing schemes which was introduced by Pieprzyk and Zhang in [94]. When a group of participants submit invalid shares to a combiner, their main goals are to learn the secret and prevent the honest ones from detecting the fact of cheating. In a cheating-immune scheme, the cheaters will not gain any advantage over the honest ones in learning the secret.

Note that the approach adopted by Pieprzyk and Zhang is different from those

used in code-based secret sharing schemes, VSSS and SSS with cheating detection or identification. There is no detection/identification of invalid shares in a cheating-immune scheme, it only prevents the cheaters from knowing the secret if the honest ones are unable to do so. However, in a cheating-immune scheme, the size of the shares can be of the same size as the secret and there is no need for extra shares during reconstruction.

There are only a few known ways to construct cheating-immune (n, n) -secret sharing schemes over $\mathbb{F}_2$ [93,94]. In this work, we revisit two methods to construct Boolean functions satisfying multiple cryptographic criteria. These methods involve the use of generator matrices of binary linear codes. By adding some conditions, we prove that these methods can be used to build new cheating-immune (n, n) -secret sharing schemes. We remark that the method we used to construct secret sharing schemes from codes is different from those used in the well-known code-based secret sharing schemes [67, 83, 85].

The only known cheating-immune schemes are for the (n, n) -threshold access structure and it is an open problem to construct cheating-immune schemes for more general access structures. In connection with this problem, we study the application of two general constructions of secret sharing schemes to build cheating-immune (t, n) -threshold schemes. The first method uses cumulative arrays while the second one uses generalized cumulative arrays from perfect hash families. We analyze the parameters of the schemes obtained and we also compare the two methods. However, we note that in the resulting schemes, the size of the shares are generally large. Thus, the construction of cheating-immune (t, n) -threshold schemes with the smallest possible share size is still an open question.

1.3 Organization of this thesis

The rest of this thesis is organized as follows. In Chapter 2, we present the theoretical background for the topics and results in this work. Chapter 3 discusses the theory of secret sharing schemes. In Chapter 4, we study the access structure and multiplicativity of linear secret sharing schemes based on codes from complete graphs. Chapter 5 talks about the access structure of a generalized secret sharing scheme based on linear codes. In Chapter 6, we present a study of the maximum number of minimal codewords in binary linear codes. Chapter 7 deals with constructions of new cheating-immune secret sharing schemes. Finally, we conclude this work and present some open problems in Chapter 8.

Chapter 2

Preliminaries

We present in this chapter the theoretical background for the topics considered in this thesis. The reader interested in a deeper treatment of the subjects in this chapter may consult the main references.

2.1 Linear codes

An error-correcting code enables the sending of messages over a noisy communication channel such that errors made during transmission are detected and corrected. We discuss in this section the concepts in coding theory that will be used in this thesis. The main reference that we use is [59].

Let $\mathbb{F}_q$ stand for the finite field of order q , where q is a prime power. We denote by $\mathbb{F}_q^n$ the vector space of all n -tuples over $\mathbb{F}_q$. A vector $\mathbf{v} \in \mathbb{F}_q^n$ will also be written as $\mathbf{v} = (v_1, v_2, \dots, v_n)$.

Definition 2.1.1. Let $\mathbf{v} \in \mathbb{F}_q^n$. The support of $\mathbf{v}$ is given by

$$\text{supp}(\mathbf{v}) = \{i : v_i \neq 0, 1 \leq i \leq n\}.$$

The Hamming weight $\text{wt}(\mathbf{v})$ of $\mathbf{v}$ is the number of its nonzero coordinates, i.e.,

$$\text{wt}(\mathbf{v}) = |\text{supp}(\mathbf{v})|.$$

Definition 2.1.2. The distance $d(\mathbf{u}, \mathbf{v})$ between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$ is defined to be the number of positions at which $\mathbf{u}$ and $\mathbf{v}$ differ, that is,

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

Definition 2.1.3. An (n, M) code $\mathcal{C}$ over $\mathbb{F}_q$ is a subset of size M of $\mathbb{F}_q^n$. An element of $\mathcal{C}$ is called a codeword. An $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ is a k -dimensional subspace of $\mathbb{F}_q^n$.

Definition 2.1.4. The minimum distance of $\mathcal{C}$, denoted by $d(\mathcal{C})$, is given by

$$d(\mathcal{C}) = \min\{d(\mathbf{u}, \mathbf{v}) : \mathbf{u}, \mathbf{v} \in \mathcal{C}, \mathbf{u} \neq \mathbf{v}\}.$$

The minimum distance of a linear code $\mathcal{C}$ is equal to the smallest of the Hamming weights of the nonzero codewords of $\mathcal{C}$ (also called the *minimum Hamming weight* of $\mathcal{C}$). For a fixed n and k , the higher the minimum distance, the more errors that can be corrected by the code. A linear code $\mathcal{C}$ over $\mathbb{F}_q$ of length n , dimension k , and minimum distance d is called an $[n, k, d]_q$ linear code or an $[n, k, d]$ linear code.

Let $\mathcal{C}$ be an (n, M) code over $\mathbb{F}_q$. We define the *information rate* or *rate* of $\mathcal{C}$ as $n^{-1} \log_q M$. If $\mathcal{C}$ is an $[n, k]$ linear code over $\mathbb{F}_q$ then $n^{-1} \log_q M = k/n$. The rate is a measure of the number of coordinates which contain information relative to the total number of coordinates.

We say that two linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are *equivalent* if one can be obtained from the other by a series of operations of the following types: (i) permutation of the n coordinates; and (ii) multiplication by a nonzero scalar of the entries in a fixed coordinate position.

A *subcode* $\mathcal{D}$ of a linear code $\mathcal{C}$ is a subspace of $\mathcal{C}$. The *support* of a subcode $\mathcal{D}$, denoted by $\text{supp}(\mathcal{D})$, is the union of the supports of its codewords.

Definition 2.1.5. For $1 \leq l \leq k$, the l th generalized Hamming weight of $\mathcal{C}$ is given by

$$d_l(\mathcal{C}) = d_l = \min\{|\text{supp}(\mathcal{D})| : \mathcal{D} \text{ is an } [n, l] \text{ subcode of } \mathcal{C}\}.$$

A linear code is usually represented by its *generator matrix* or its *parity-check matrix*.

Definition 2.1.6. Let $\mathcal{C}$ be an $[n, k]$ linear code. A generator matrix G for $\mathcal{C}$ is a $k \times n$ matrix whose rows form a basis for $\mathcal{C}$. A parity-check matrix H for $\mathcal{C}$ is an $n - k \times n$ matrix satisfying the following property

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n : H\mathbf{c}^T = \mathbf{0}^T\}.$$

If G is a generator matrix for an $[n, k]$ linear code $\mathcal{C}$ then

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n : \mathbf{c} = \mathbf{u}G \text{ for some } \mathbf{u} \in \mathbb{F}_q^k\}.$$

For a codeword $\mathbf{c} \in \mathcal{C}$, the vector $\mathbf{u} \in \mathbb{F}_q^k$ such that $\mathbf{c} = \mathbf{u}G$ is called an *information vector*. In general, a linear code has many possible generator matrices and parity-check matrices. A generator matrix G is said to be in *standard form* if $G = [I_k \mid A]$ where I_k is the $k \times k$ identity matrix. A linear code $\mathcal{C}$ has a unique generator matrix

in standard form (up to equivalence). Given a generator matrix G for $\mathcal{C}$ in standard form, then $H = [-A^T \mid I_{n-k}]$ is a parity-check matrix for $\mathcal{C}$.

Let $\mathbf{u} \in \mathbb{F}_q^k$ be an information vector and consider the corresponding codeword $\mathbf{c} = \mathbf{u}G$ where G is a generator matrix in standard form. Then we can see that the first k coordinates of $\mathbf{c}$ is the vector $\mathbf{u}$. In this case, the first k coordinates is said to form an *information set* for $\mathcal{C}$ while the remaining $n - k$ coordinates form a *redundancy set*.

Example 2.1.7. *The binary repetition code of length n is the $[n, 1, n]$ code $\mathcal{C}$ consisting of two codewords $(00 \dots 0)$ and $(11 \dots 1)$. Suppose we sent a codeword of $\mathcal{C}$ through a channel and e errors were introduced during transmission. We can see that if $1 \leq e \leq n - 1$ then the received vector is not a valid codeword. This means that $\mathcal{C}$ can detect at most $n - 1$ errors. If $1 \leq e \leq \lfloor (n - 1)/2 \rfloor$ then we can deduce that more than half of the coordinates of the received vector are correct. Hence, we can correct the errors in the received vector. A generator matrix for $\mathcal{C}$ is given by*

$$G = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$$

and a parity-check matrix is

$$H = \left[\begin{array}{c|c} 1 & I_{n-1} \\ \vdots & \\ 1 & \end{array} \right]$$

where I_{n-1} is the $(n - 1) \times (n - 1)$ identity matrix.

In general, if a code $\mathcal{C}$ has minimum distance d then it can detect at most $d - 1$ errors and it can correct at most $\lfloor (d - 1)/2 \rfloor$ errors.

Let $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$. We denote by $\langle \mathbf{u}, \mathbf{v} \rangle$ the standard inner product of $\mathbf{u}$ and $\mathbf{v}$.

Definition 2.1.8. The dual code $\mathcal{C}^\perp$ of a linear code $\mathcal{C}$ is defined as

$$\mathcal{C}^\perp = \{\mathbf{v} \in \mathbb{F}_q^n : \langle \mathbf{v}, \mathbf{c} \rangle = 0, \forall \mathbf{c} \in \mathcal{C}\}.$$

Let $\mathcal{C}$ be an $[n, k]$ linear code. It follows from linear algebra that $\mathcal{C}^\perp$ is an $[n, n - k]$ linear code. We use the notation $d^\perp$ for the minimum distance of $\mathcal{C}^\perp$ (also called the *dual distance* of $\mathcal{C}$). Suppose $\mathcal{C}$ has a generator matrix G and parity-check matrix H . Then H is a generator matrix and G is a parity-check matrix for $\mathcal{C}^\perp$.

Definition 2.1.9. A linear code $\mathcal{C}$ is said to be *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$ and it is *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$.

The dimension k of a self-orthogonal code is at most $n/2$. If an $[n, k]$ code is self-dual then n must be even and $k = n/2$.

Definition 2.1.10. Let $A_i(\mathcal{C})$, or simply A_i , be the number of codewords of Hamming weight i in a linear code $\mathcal{C}$. Then the *weight distribution* or *weight spectrum* of $\mathcal{C}$ is the set of numbers $\{A_0, A_1, A_2, \dots, A_n\}$.

When giving the weight distribution of a linear code $\mathcal{C}$, only the nonzero A_i are listed. In general, it is a very difficult problem to determine the weight distribution of linear codes [15]. Some basic results about the weight distribution are given in the next theorem.

Theorem 2.1.11. Let $\mathcal{C}$ be an $[n, k, d]$ linear code over $\mathbb{F}_q$. Then:

i. $A_0 = 1$ and for $i = 1, 2, \dots, d - 1$, we have $A_i = 0$.

ii. $A_0 + A_1 + \dots + A_n = |\mathcal{C}| = q^k$.

iii. If $\mathcal{C}$ is binary and contains the all-one codeword $(1, 1, \dots, 1)$ then $A_i = A_{n-i}$ for

$$0 \leq i \leq n.$$

iv. If $\mathcal{C}$ is binary and self-orthogonal then each codeword has even weight.

v. If $\mathcal{C}$ is ternary and self-orthogonal then the weight of each codeword is divisible by three.

We now define the *weight enumerator* of a linear code $\mathcal{C}$. It is the generating polynomial of the weight distribution.

Definition 2.1.12. *The weight enumerator of a linear code $\mathcal{C}$ is the polynomial*

$$W_{\mathcal{C}}(x) = \sum_{i=0}^n A_i(\mathcal{C})x^i$$

or the homogeneous polynomial

$$W_{\mathcal{C}}(x, y) = \sum_{i=0}^n A_i(\mathcal{C})x^i y^{n-i}.$$

The weight distribution of $\mathcal{C}$ can be uniquely determined by the weight distribution of the dual code $\mathcal{C}^\perp$. The next result is the well-known MacWilliams identity.

Theorem 2.1.13. *[59, Section 7.2]*

$$W_{\mathcal{C}^\perp}(x, y) = \frac{1}{|\mathcal{C}|} W_{\mathcal{C}}(y - x, y + (q - 1)x)$$

Definition 2.1.14. *Let $\mathcal{C}$ be an $[n, k, d]$ linear code over $\mathbb{F}_q$. The covering radius $\rho(\mathcal{C})$ of $\mathcal{C}$ is defined as*

$$\rho(\mathcal{C}) = \max_{\mathbf{u} \in \mathbb{F}_q^n} \min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{u}, \mathbf{c}).$$

The covering radius can be interpreted as the smallest integer ρ such that every vector in $\mathbb{F}_q^n$ has distance ρ or less from at least one codeword in $\mathcal{C}$. It is used in knowing the error-correcting capability of a code.

We next present several ways of constructing codes from other codes.

Definition 2.1.15. *Let $\mathcal{C}$ be an $[n, k, d]$ linear code over $\mathbb{F}_q$ and let $J \subseteq \{1, \dots, n\}$.*

- i. The code punctured on J , denoted by $\mathcal{C}^J$, is defined as the code obtained by deleting components indexed by the set J in all codewords of $\mathcal{C}$.*
- ii. Let $\mathcal{C}(J)$ be the subcode of $\mathcal{C}$ consisting of codewords that are 0 on J . Then the code shortened on J , denoted by $\mathcal{C}_J$, is defined as the code obtained by puncturing $\mathcal{C}(J)$ on J .*
- iii. The extended code $\widehat{\mathcal{C}}$ is given by*

$$\widehat{\mathcal{C}} = \{(c_1, \dots, c_n, c_{n+1}) \in \mathbb{F}_q^{n+1} : (c_1, \dots, c_n) \in \mathcal{C} \text{ and } c_1 + \dots + c_n + c_{n+1} = 0\}.$$

The punctured code $\mathcal{C}^J$ is an $[n - |J|, k', d']$ code where $k - |J| \leq k' \leq k$ and minimum distance $d' \geq d - |J|$. The shortened code $\mathcal{C}_J$ is an $[n - |J|, k'', d'']$ code where $k - |J| \leq k'' \leq k$ and minimum distance $d'' \geq d$. We have the following relations between the punctured and shortened codes of $\mathcal{C}$ and its dual $\mathcal{C}^\perp$ [59, Section 1.5]:

$$(\mathcal{C}^\perp)_J = (\mathcal{C}^J)^\perp \text{ and } (\mathcal{C}^\perp)^J = (\mathcal{C}_J)^\perp.$$

The extended code $\widehat{\mathcal{C}}$ is an $[n + 1, k]$ code with minimum distance d or $d + 1$. Let G be a generator matrix for $\mathcal{C}$. Then a generator matrix $\widehat{G}$ for $\widehat{\mathcal{C}}$ is obtained by adding a column to G such that the sum of coordinates of each row of $\widehat{G}$ is zero. Let H be

a parity-check matrix for $\mathcal{C}$. Then a parity-check matrix $\widehat{H}$ for $\widehat{\mathcal{C}}$ is given by

$$\widehat{H} = \left[\begin{array}{ccc|c} 1 & \dots & 1 & 1 \\ \hline & & & 0 \\ & H & & \vdots \\ & & & 0 \end{array} \right].$$

Definition 2.1.16. Let $\mathcal{C}_i$ be an $[n_i, k_i, d_i]$ linear code over $\mathbb{F}_q$ for $i = 1, 2$. The direct sum of $\mathcal{C}_1$ and $\mathcal{C}_2$ is given by

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(\mathbf{c}_1, \mathbf{c}_2) : \mathbf{c}_1 \in \mathcal{C}_1, \mathbf{c}_2 \in \mathcal{C}_2\}.$$

The direct sum $\mathcal{C}_1 \oplus \mathcal{C}_2$ is an $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]$ linear code. For $i = 1, 2$, suppose $\mathcal{C}_i$ has generator matrix G_i and parity-check matrix H_i then $\mathcal{C}_1 \oplus \mathcal{C}_2$ has generator matrix and parity-check matrix

$$G_1 \oplus G_2 = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}, H_1 \oplus H_2 = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}.$$

Definition 2.1.17. Let r, n be integers such that $r \geq 2$ and $n = 2^r - 1$. Consider the $r \times (2^r - 1)$ matrix H_r whose columns are exactly all the nonzero vectors of $\mathbb{F}_2^r$. The binary Hamming code of length $n = 2^r - 1$, denoted by $\text{Ham}_{2,r}$, is defined as the linear code with parity-check matrix H_r .

Note that if we fix an ordering of the columns in the definition, then any reordering of the columns results into an equivalent code. It follows that the definition refers to a class of equivalent codes. The binary Hamming code $\text{Ham}_{2,r}$ is a $[2^r - 1, 2^r - 1 - r, 3]$ linear code. Since H_r contains all the vectors in $\mathbb{F}_2^r$ of weight one, then it follows that the rows of H_r are linearly independent. Hence, the dimension of

$Ham_{2,r}$ is $2^r - 1 - r$. To determine the minimum distance d , first we note that the columns of H_r are distinct and nonzero. This implies that $d \geq 3$. Now the columns $(1, 0, \dots, 0)^T, (0, 1, 0, \dots, 0)^T, (1, 1, 0, \dots, 0)^T$ are linearly dependent. Thus, $d = 3$.

Definition 2.1.18. *The dual of the binary Hamming code $Ham_{2,r}$ is called the binary simplex code $S_{r,2}$.*

The binary simplex code $S_{r,2}$ has length $2^r - 1$, dimension r , and minimum distance 2^{r-1} . The nonzero codewords of $S_{r,2}$ all have weights 2^{r-1} .

Example 2.1.19. *Consider the $[7, 4, 3]$ binary Hamming code $Ham_{2,3}$. A generator matrix for $Ham_{2,3}$ is*

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

The extended code obtained from $Ham_{2,3}$ is the $[8, 4, 4]$ extended binary Hamming code which is a self-dual code.

Let $A_q(n, d)$ be the maximum number of codewords in a code (linear or nonlinear) over $\mathbb{F}_q$ of length n and minimum distance d . The next theorem is the *Singleton bound*.

Theorem 2.1.20. *[59, Section 2.4] For $d \leq n$, we have $A_q(n, d) \leq q^{n-d+1}$. If an $[n, k, d]$ linear code over $\mathbb{F}_q$ exists then $k \leq n - d + 1$.*

A linear code $\mathcal{C}$ is said to be *maximum distance separable (MDS)* if equality is achieved in the Singleton bound for this code. Suppose an $[n, k, d]$ linear code $\mathcal{C}$ is

MDS. Then there does not exist another $[n, k]$ code with a larger minimum distance than $\mathcal{C}$. Equivalently, there does not exist another code of length n and distance d with more codewords than $\mathcal{C}$. The next theorem gives some properties of MDS codes.

Theorem 2.1.21. *[59, Section 2.4] Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$ and suppose G and H are generator and parity-check matrices, respectively, for $\mathcal{C}$. Then we have the following equivalent statements.*

- i. $\mathcal{C}$ is MDS.*
- ii. Every set of k columns of G is linearly independent.*
- iii. $\mathcal{C}^\perp$ is MDS.*
- iv. Every set of $n - k$ columns of H is linearly independent.*

Definition 2.1.22. *Let n, k be integers such that $1 \leq n \leq q$ and $1 \leq k \leq n$. Let $x_0, x_1, \dots, x_{n-1}$ be n distinct elements of $\mathbb{F}_q$. The Reed-Solomon code $\mathcal{C}$ is defined by*

$$\mathcal{C} = \{(f(x_0), f(x_1), \dots, f(x_{n-1})) : f(x) \in \mathbb{F}_q[x], \deg(f) < k\}$$

where $\deg(f)$ is the degree of the polynomial $f(x)$.

The Reed-Solomon code is an $[n, k]$ linear code over $\mathbb{F}_q$. Since a nonzero polynomial $f(x)$ of degree less than k can have at most $k - 1$ zeros then it follows that $d(\mathcal{C}) \geq n - k + 1$. By the Singleton bound, $d(\mathcal{C}) \leq n - k + 1$. Thus, the minimum distance of the Reed-Solomon code is equal to $n - k + 1$. Therefore, the Reed-Solomon code is an MDS code.

We now discuss the concept of minimal codewords.

Definition 2.1.23. For any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$, we say that $\mathbf{v}$ covers $\mathbf{u}$, denoted by $\mathbf{u} \preceq \mathbf{v}$, if $\text{supp}(\mathbf{u}) \subseteq \text{supp}(\mathbf{v})$. When $\text{supp}(\mathbf{u}) \subset \text{supp}(\mathbf{v})$ we use the notation $\mathbf{u} \prec \mathbf{v}$.

For example, let $\mathbf{v} = (1, 1, 1, 1), \mathbf{u} = (1, 0, 1, 0) \in \mathbb{F}_2^4$. Then $\mathbf{v}$ covers $\mathbf{u}$.

Definition 2.1.24. A nonzero codeword of a linear code $\mathcal{C}$ is called minimal if it covers only its scalar multiples and no other nonzero codewords.

Alternatively, a nonzero codeword $\mathbf{c}$ is minimal if there does not exist a nonzero codeword $\mathbf{u}$ such that $\text{supp}(\mathbf{u}) \subset \text{supp}(\mathbf{c})$. We can see that given a minimal codeword $\mathbf{c}$ then its scalar multiples are also minimal. Let $\mathcal{W}(\mathcal{C})$ be the set of minimal codewords of a linear code $\mathcal{C}$ and let $W(\mathcal{C}) = |\mathcal{W}(\mathcal{C})|$.

Minimal codewords were introduced by Hwang [60] in relation to a decoding algorithm whose complexity depends on the set of minimal codewords. The study of minimal codewords was continued by Agrell in [1, 2]. He investigated the problem of determining the minimal codewords of binary linear codes and presented an asymptotic analysis of the ratio of the number minimal codewords to the total number of codewords in long binary codes. Massey re-introduced the concept of minimal codewords in connection with secret sharing [83, 84]. He showed that the set of minimal codewords determines the access structure of the secret sharing schemes based on linear codes. In general, it is very difficult to determine the minimal codewords of a linear code.

The following lemma in [5] states some properties of minimal codewords.

Lemma 2.1.25. Let $\mathcal{C}$ be an $[n, k, d]$ linear code over $\mathbb{F}_q$ and suppose H is a parity-check matrix of $\mathcal{C}$.

i. Let $\mathbf{c}$ be a codeword of $\mathcal{C}$ with $\text{supp}(\mathbf{c}) = U$. Then $\mathbf{c}$ is minimal if and only if the rank of $H(U)$ is equal to $|U| - 1$ where $H(U)$ is the restriction of H to columns indexed by U .

ii. If $\mathbf{c}$ is a minimal codeword then $\text{wt}(\mathbf{c}) \leq n - k + 1$.

iii. If the Hamming weight of a codeword $\mathbf{c}$ satisfies $\text{wt}(\mathbf{c}) \leq d(1 + 1/(q - 1)) - 1$ then $\mathbf{c}$ is minimal.

iv. The linear span of $\mathcal{W}(\mathcal{C})$ is the linear code $\mathcal{C}$.

v. Suppose $\mathbf{c}$ is a non-minimal codeword of a binary code $\mathcal{C}$. Then there exist two codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}$ with disjoint supports such that $\mathbf{u}, \mathbf{v} \prec \mathbf{c}$ and $\mathbf{c} = \mathbf{u} + \mathbf{v}$.

Statement (i) of Lemma 2.1.25 already appeared in [2] as the “H rule”. Moreover, in the same paper, we have the following “G rule”.

Lemma 2.1.26. *Let $\mathcal{C}$ be an $[n, k, d]$ linear code over $\mathbb{F}_q$ and suppose G is a generator matrix for $\mathcal{C}$. Let $\mathbf{c}$ be a codeword of $\mathcal{C}$ and define $V = \{1, 2, \dots, n\} \setminus \text{supp}(\mathbf{c})$. Then $\mathbf{c}$ is minimal if and only if the rank of $G(V)$ is equal to $k - 1$ where $G(V)$ is the restriction of G to columns indexed by V .*

Example 2.1.27. *Consider the binary $[6, 3, 1]$ linear code $\mathcal{C}$ with generator matrix*

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

The codewords of $\mathcal{C}$ are

$(000000), (110000), (111000), (001000), (001011), (111011), (110011), (000011).$

The set of minimal codewords of $\mathcal{C}$ is

$$\mathcal{W}(\mathcal{C}) = \{(001000), (110000), (000011)\}.$$

Example 2.1.28. Let $\mathcal{C}$ be an $[n, k, d]$ MDS code over $\mathbb{F}_q$. By Lemma 2.1.25 (ii), the only minimal codewords of $\mathcal{C}$ are the codewords of minimum weight. We can deduce from Theorem 2.1.21 that the number of codewords of minimum weight is equal to $(q - 1)\binom{n}{d}$.

2.2 Graphs

The definitions and results in this section are taken from Sections 1.1, 1.2 and 8.6 of [114]. Let $G(V, E)$ be an undirected graph with vertex set $V = \{v_1, \dots, v_m\}$ and edge set $E = \{e_1, \dots, e_n\}$. An edge e with endpoints v_i and v_j will also be denoted by $v_i v_j$. We say that two vertices are *adjacent* if they are the endpoints of an edge. The *complete* graph K_m on m vertices is the graph with every pair of vertices adjacent. Thus, K_m has $\binom{m}{2}$ edges.

A *path* from v_0 to v_l in G is a sequence of vertices $v_0, v_1, \dots, v_l$ such that the set E contains the edges $v_0 v_1, v_1 v_2, \dots, v_{l-1} v_l$. A graph G is *connected* if for every pair of vertices v_i, v_j , there is a path from v_i to v_j . We only consider connected graphs in this thesis. A *cycle* is a path with at least three vertices and with the same start and end vertices. An *odd* (resp. *even*) *cycle* is a cycle with an odd (resp. even) number of vertices. A *cycle graph with m vertices* is the graph C_m that is a cycle. A *cut-set* is a minimal set of edges such that removing these edges will make the graph disconnected.

A *subgraph* $G'(V', E')$ of $G(V, E)$ is a graph with $V' \subseteq V$ and $E' \subseteq E$. For a subset $E' \subseteq E$, we define the *edge-induced subgraph* $G[E']$ as the subgraph with edge set E' and with vertex set consisting of the endpoints of the edges in E' . For each subgraph G' of G , we can associate a binary characteristic vector $\mathbf{g} = (g_1, \dots, g_n)$, where $g_i = 1$ if e_i is an edge of G' and $g_i = 0$ otherwise, $1 \leq i \leq n$. With this association, the set of all subgraphs forms a vector space $\mathcal{W}(G)$ over $\mathbb{F}_2$.

We consider two subspaces of $\mathcal{W}(G)$. The first one is the *cut-set space* (or *bond space*) of G , denoted by $\mathcal{C}(G)$, consisting of the characteristic vectors of cut-sets and union of edge-disjoint cut-sets. The dimension of $\mathcal{C}(G)$ is equal to $m - 1$. The second subspace is the *cycle space* $\mathcal{C}^*(G)$ of G . It consists of the cycles and the union of edge-disjoint cycles. The subspaces $\mathcal{C}(G)$ and $\mathcal{C}^*(G)$ are orthogonal to each other with respect to the inner product in the binary field. It follows that the dimension of $\mathcal{C}^*(G)$ is equal to $n - m + 1$. As binary linear codes, $\mathcal{C}(G)$ is an $[n, m - 1]$ code and $\mathcal{C}^*(G)$ is an $[n, n - m + 1]$ code. The minimum distance of $\mathcal{C}(G)$ is equal to the number of edges in the smallest cut-set while the minimum distance of $\mathcal{C}^*(G)$ is equal to the number of edges in the smallest cycle (this number is also called the *girth* of the graph). Moreover, the dual code of $\mathcal{C}(G)$ is $\mathcal{C}^*(G)$. We will also call $\mathcal{C}(G)$ and $\mathcal{C}^*(G)$ as the *cut-set code* and the *cycle code* of G , respectively.

Example 2.2.1. Consider the complete graph K_m with m vertices. Then the cut-set code $\mathcal{C}(G)$ has parameters $[n, m - 1, m - 1]$ and the cycle code $\mathcal{C}^*(K_m)$ has parameters $[n, n - m + 1, 3]$ where $n = \binom{m}{2}$.

The *incidence matrix* $H = (h_{ij})_{m \times n}$ of G is a matrix over $\mathbb{F}_2$ defined by

$$h_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases}.$$

The *incidence set* of a vertex v_i in G is the cut-set consisting of the set of edges that are incident with v . This set can be represented by the row in the incidence matrix H which corresponds to the vertex v_i (called the *incidence vector* of v_i). The incidence vectors of any $m - 1$ vertices of a connected m -vertex graph G form a basis of the cut-set space $\mathcal{C}(G)$.

For $i = 1, \dots, n$, let H_i be the submatrix of H formed by removing the incidence vector of v_i . We call H_i a *reduced incidence matrix*. Without loss of generality, we fix removing the vertex v_m and then always use the notation H_m since any transformation of the vertex set yields an isomorphic graph.

A *bipartite graph* is a graph with the property that the vertex set can be divided into two disjoint sets V_1 and V_2 such that for every edge, one of its endpoints is in V_1 while the other is in V_2 . A bipartite graph can be characterized in terms of its cycles.

Lemma 2.2.2. *A graph is bipartite if and only if it does not contain an odd cycle.*

2.3 Matroids

The material here on matroid theory is taken from Sections 1.1, 1.2, 1.3 and 10.1 of [90].

Definition 2.3.1. *Let Q be a finite set and let $\mathcal{I}$ be a collection of subsets of Q . A matroid M is a pair $(Q, \mathcal{I})$ satisfying the following three conditions:*

1. $\emptyset \in \mathcal{I}$.
2. If $X \in \mathcal{I}$ and $Y \subseteq X$ then $Y \in \mathcal{I}$.
3. If $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then there exists an element y of $Y \setminus X$ such that $X \cup \{y\} \in \mathcal{I}$.

The subsets $X \subseteq Q$ that are in $\mathcal{I}$ are said to be *independent*. A *basis* of a matroid is a maximal independent set. The *dependent* sets are those that are not independent, and a *circuit* is a minimal dependent set. There is an equivalent definition of a matroid in terms of its circuits. A matroid M is said to be *connected* if, for every two points in Q (which is called the *ground set*), there exists a circuit containing them.

For a subset $X \subseteq Q$, we define the *rank of X* or $r(X)$ to be equal to the cardinality of the maximal independent subset of X . The rank of a matroid M is defined to be the rank of Q . The rank function of M is the function which maps every subset $X \subseteq Q$ to its rank $r(X)$. A matroid can also be defined by the rank function.

All bases have the same number of elements and this number is the rank of M . Let B be a base of M and $x \in Q \setminus B$. Then there exists a circuit C such that $x \in C \subseteq B \cup x$. This circuit C is called the *fundamental circuit of x in B* .

We give two well-known examples of matroids.

Example 2.3.2. Let A be an $m \times n$ matrix over a finite field $\mathbb{F}_q$. We label the columns by $1, \dots, n$ and let $Q = \{1, \dots, n\}$. Define $\mathcal{I}$ to be the collection of subsets X of Q such that the set of columns labelled by X are linearly independent. Then $M = (Q, \mathcal{I})$ is a matroid. We also denote this matroid by $M[A]$.

Example 2.3.3. Let Q be a set with n elements and define $\mathcal{I}$ as

$$\mathcal{I} = \{X \subseteq Q : |X| \leq k\}$$

where $k \leq n$. Then $M = (Q, \mathcal{I})$ is a matroid called the *uniform matroid of rank k* and is denoted by $U_{k,n}$. The bases of M are all the subsets of Q of size k while the circuits of M are all the subsets of Q of size $k + 1$.

Two matroids $M_1 = (Q_1, \mathcal{I}_1), M_2 = (Q_2, \mathcal{I}_2)$ are said to be *isomorphic* if there exists a bijection $\phi : Q_1 \rightarrow Q_2$ such that for all $X \subseteq Q_1$, $\phi(X)$ is independent in M_2 if and only if X is independent in M_1 . A matroid $M = (Q, \mathcal{I})$ is said to be *representable over $\mathbb{F}_q$* (or $\mathbb{F}_q$ -representable) if there exists a matrix A over $\mathbb{F}_q$ such that M is isomorphic to $M[A]$. A *binary matroid* is one that is representable over $\mathbb{F}_2$. A matroid M on a ground set with n elements and of rank k is said to be *uniquely $\mathbb{F}_q$ -representable* if all of the $k \times n$ matrices representing M over $\mathbb{F}_q$ are equivalent. The characterization of representable matroids is one of the main open problems in matroid theory. We will need the following well-known results on binary matroids.

Lemma 2.3.4. *A binary matroid is uniquely $\mathbb{F}_2$ -representable.*

Lemma 2.3.5. *If a binary matroid is representable over a field $\mathbb{F}_q$, then it is uniquely $\mathbb{F}_q$ -representable.*

Example 2.3.6. Let G be an undirected graph. The cycle matroid associated with G is the matroid $M(G)$ on the edge set $E(G)$ whose circuits are the cycles of G . As a binary matroid, $M(G)$ can be represented by the vertex-edge incidence matrix H of G . A matroid is said to be *graphic* if it is isomorphic to the cycle matroid of some graph.

Theorem 2.3.7. *Let M be a matroid on a set Q and let $\mathcal{B}$ be the set of bases of M . Consider the set $\mathcal{B}^* = \{Q \setminus B : B \in \mathcal{B}\}$. Then $\mathcal{B}^*$ is the set of bases of a matroid M^* on Q .*

The matroid M^* is called the *dual* of M . The bases and circuits of M^* are called the *cobases* and *cocircuits* of M . If M is representable over $\mathbb{F}_q$ then so is M^* .

Example 2.3.8. *Consider the uniform matroid $U_{k,n}$ of rank k . Applying Theorem 2.3.7, the set $\mathcal{B}^*$ will consist of all subsets of size $n - k$. Hence, the dual of $U_{k,n}$ is isomorphic to $U_{n-k,n}$.*

Example 2.3.9. *The dual of the cycle matroid $M(G)$ of an undirected graph G is the matroid $M^*(G)$ whose circuits are the set of edges of the cut-sets of G . The matroid $M^*(G)$ is called the *cut-set matroid* of G . A matroid is said to be *cographic* if it is isomorphic to $M^*(G)$.*

Given an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ then there is a unique matroid $M(\mathcal{C})$ associated with it. Any generator matrix of $\mathcal{C}$ is a representation over $\mathbb{F}_q$ of the matroid $M(\mathcal{C})$. We note that a representable matroid can be associated with different codes. The dual matroid of $M(\mathcal{C})$ is the matroid $M(\mathcal{C}^\perp)$ associated with the dual code of $\mathcal{C}$, i.e. $M^*(\mathcal{C}) \cong M(\mathcal{C}^\perp)$. A dependent set of $M(\mathcal{C})$ corresponds to a codeword in $\mathcal{C}^\perp$. It follows that a circuit of $M(\mathcal{C})$ corresponds to a minimal codeword of $\mathcal{C}^\perp$.

Example 2.3.10. *Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$ and suppose that $\mathcal{C}$ is MDS. Then the matroid $M(\mathcal{C})$ is isomorphic to the uniform matroid $U_{k,n}$. Similarly, the matroid $M(\mathcal{C}^\perp)$ is isomorphic to $U_{n-k,n}$.*

Example 2.3.11. *Let G be a connected graph with m vertices and n edges. The matroid associated with the cut-set code $\mathcal{C}(G)$ has as its circuits the cycles of G .*

Hence, $M(\mathcal{C}(G))$ is isomorphic to the cycle matroid $M(G)$. The matroid associated with the cycle code $\mathcal{C}^*(G)$ is isomorphic to the cut-set matroid $M^*(G)$.

2.4 Entropy

We present some of the basic concepts related to the entropy of a random variable. These concepts will be used in the definition of a secret sharing scheme. Our reference for this section are Sections 2.1 and 2.2 of [35].

We denote a discrete random variable taking values on a set X by $\mathbf{X}$. The probability mass function for a random variable $\mathbf{X}$ will be denoted by $p_{\mathbf{X}}(x) = \Pr[\mathbf{X} = x]$ for $x \in X$.

Definition 2.4.1. Let $\mathbf{X}$ be a discrete random variable with probability mass function $p_{\mathbf{X}}(x)$. The entropy $H(\mathbf{X})$ is defined by

$$H(\mathbf{X}) = - \sum_{x \in X} p_{\mathbf{X}}(x) \log p_{\mathbf{X}}(x).$$

In the definition, the logarithm is to the base 2. This implies that the entropy is expressed in bits. When $p_{\mathbf{X}}(x) = 0$ we follow the convention that $p_{\mathbf{X}}(x) \log p_{\mathbf{X}}(x) = 0$. The entropy is a measure of the amount of uncertainty in a random variable. We can also say that it measures the amount of information in a random variable. The definition can be extended to a pair of random variables $\mathbf{X}, \mathbf{Y}$ using the corresponding joint probability distribution. We use the notation $H(\mathbf{XY})$ for the joint entropy of the random variables $\mathbf{X}$ and $\mathbf{Y}$.

Example 2.4.2. Let $\mathbf{X}$ be a random variable such that it is equal to 1 with probability

p and it is equal to 0 with probability $1 - p$. Then

$$H(\mathbf{X}) = -p \log p - (1 - p) \log(1 - p).$$

In particular, if $p = 1/2$ then $H(\mathbf{X}) = 1$. Ranging through all possible values of p , we can define a function

$$H(p) = -p \log p - (1 - p) \log(1 - p).$$

This function is called the Shannon's binary entropy function.

Definition 2.4.3. Let $\mathbf{X}$ and $\mathbf{Y}$ be discrete random variables with probability mass functions $p_{\mathbf{X}}(x)$ and $p_{\mathbf{Y}}(y)$ respectively. Then we have the following:

$$i. H(\mathbf{Y}|\mathbf{X} = x) = - \sum_{y \in Y} p_{\mathbf{Y}|\mathbf{X}}(y|x) \log p_{\mathbf{Y}|\mathbf{X}}(y|x).$$

ii. The conditional entropy $H(\mathbf{Y}|\mathbf{X})$ is defined as

$$H(\mathbf{Y}|\mathbf{X}) = \sum_{x \in X} p_{\mathbf{X}}(x) H(\mathbf{Y}|\mathbf{X} = x).$$

We list down some properties of the entropy:

1. $0 \leq H(\mathbf{X}) \leq \log |X|$
2. If $\mathbf{X}$ is uniformly distributed over X then $H(\mathbf{X}) = \log |X|$.
3. $H(\mathbf{YX}) = H(\mathbf{Y}|\mathbf{X}) + H(\mathbf{X})$
4. $0 \leq H(\mathbf{Y}|\mathbf{X}) \leq H(\mathbf{Y})$
5. $H(\mathbf{Y}|\mathbf{X}) = H(\mathbf{Y})$ if and only if $\mathbf{Y}$ and $\mathbf{X}$ are independent.
6. $H(\mathbf{Y}|\mathbf{X}) = 0$ if and only if the value of $\mathbf{X}$ determines the value of $\mathbf{Y}$.

Definition 2.4.4. *The mutual information $I(\mathbf{X}; \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ is defined to be*

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}).$$

The mutual information is the decrease in the uncertainty of $\mathbf{X}$ due to the knowledge of $\mathbf{Y}$. We note that $I(\mathbf{X}; \mathbf{Y}) \geq 0$ and $I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{Y}; \mathbf{X})$. The mutual information can also be expressed as

$$I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}) - H(\mathbf{XY}).$$

Definition 2.4.5. *The conditional mutual information of $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ is defined as*

$$I(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{YZ}).$$

The conditional mutual information $I(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ is also nonnegative and

$$I(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) = I(\mathbf{Y}; \mathbf{X}|\mathbf{Z}).$$

2.5 Summary

In this chapter, we presented the theoretical background of the results in this thesis. These mathematical concepts include linear codes, graphs, matroids and entropy of random variables.

Chapter 3

Secret Sharing Schemes

We discuss in this chapter the theory of secret sharing schemes. We define a secret sharing scheme using information theory and then we describe its connections to matroids, monotone span programs and linear codes.

3.1 Definition

Let $\mathcal{P} = \{P_1, \dots, P_n\}$ denote the set of n participants and let D be the dealer. There are two families of subsets of $\mathcal{P}$ that are associated with a secret sharing scheme.

Definition 3.1.1. *The collection Γ of subsets of $\mathcal{P}$ that can reconstruct the secret is called the access structure.*

The elements of Γ are called *authorized sets*. The set Γ satisfies the *monotone increasing property*, that is, if $A \in \Gamma$ and $A \subseteq B$, then $B \in \Gamma$. It follows that Γ is determined by the *minimum access structure*

$$\Gamma^- = \{A \in \Gamma : \forall B \subset A \Rightarrow B \notin \Gamma\}$$

consisting of the minimal authorized sets. We only consider the case wherein every participant belongs to at least one minimal authorized set.

Definition 3.1.2. *The collection Δ of subsets of $\mathcal{P}$ that cannot obtain any information about the secret is called the privacy structure.*

The elements of Δ are called *forbidden sets*. The set Δ is *monotone decreasing*, that is, for any $B \in \Delta$, $A \subseteq B$ implies $A \in \Delta$. Hence, Δ can be described by the set

$$\Delta^+ = \{A \in \Delta : \forall B \supset A \Rightarrow B \notin \Delta\}$$

consisting of the maximal forbidden sets.

We can see that Γ and Δ are disjoint. We note that it is possible that there exists a subset of $\mathcal{P}$ that is neither authorized nor forbidden.

Example 3.1.3. *Let $\Gamma = \{B \subseteq \mathcal{P} : |B| \geq t\}$ and $\Delta = \{B \subseteq \mathcal{P} : |B| < t\}$. Then Γ is the (t, n) -threshold access structure.*

We now present a definition of secret sharing scheme. Let $\mathcal{S}$ be the set of all possible secrets such that $|\mathcal{S}| \geq 2$. For each i , $1 \leq i \leq n$, let $\mathcal{S}_i$ be the set of all possible shares for participant P_i . We represent a secret sharing scheme by a probabilistic mapping (called a *distribution mapping*)

$$\Sigma : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \mathcal{S}_n$$

where $\mathcal{R}$ is the set of random strings with a probability distribution defined on it.

We also assume that there is a probability distribution defined on $\mathcal{S}$ that governs the selection of secrets by the dealer D . The distribution mapping Σ is used as follows.

To share a secret $s \in \mathcal{S}$, D first chooses a random string $r \in \mathcal{R}$ according to the

given probability distribution and then computes $\Sigma(s, r)$ to generate the shares. The shares are then transmitted privately to the corresponding participants.

Let $A \subseteq \mathcal{P}$. The probability distribution defined on $\mathcal{S}$ together with Σ induce a probability distribution on the set of shares of the participants in A . Let us denote by $\mathbf{S}$ the random variable associated with the secret and by $\mathbf{S}_A$ the random variable associated with the set of shares of A . We also use the notation $\mathbf{S}_i$ for $\mathbf{S}_{P_i}$ where $P_i \in \mathcal{P}$. For a random variable $\mathbf{X}$, the notation $H(\mathbf{X})$ means the entropy of $\mathbf{X}$.

Definition 3.1.4. *Suppose Γ is an access structure and Δ is a privacy structure on $\mathcal{P}$. Then a distribution mapping Σ is a secret sharing scheme realizing (Γ, Δ) provided the following properties are satisfied:*

- i. For every authorized set $A \in \Gamma$, $H(\mathbf{S}|\mathbf{S}_A) = 0$.*
- ii. For every forbidden set $A \in \Delta$, $H(\mathbf{S}|\mathbf{S}_A) = H(\mathbf{S})$.*

Informally speaking, property (i) states that the shares of any set $A \in \Gamma$ determines the secret while property (ii) states that the shares of any set $A \in \Delta$ do not give any information about the secret.

Let A be an unauthorized subset of participants. We consider two types of behavior by participants belonging to A . The first type is called *honest-but-curious* wherein members of A do not deviate from the protocol but attempt to obtain additional information about the secret other than what is already given by their shares. Traditionally, this type of behavior is one of the assumptions in a secret sharing scheme.

The second type is called *malicious* wherein members of A deviate from the protocol for a variety of reasons. The reason may be to obtain additional information

about the secret or to prevent the honest participants from learning the secret or both. For example, during the reconstruction stage, some participants may use corrupted shares.

Unless otherwise stated, we assume the first type of behavior. The schemes that we consider here are *unconditionally secure*, that is, the security of the scheme is independent of the amount of computing time and resources that an entity may utilize in attempting to recover the secret by unauthorized means. In addition, we only consider schemes where the secrets and shares are uniformly distributed.

Definition 3.1.5. *A secret sharing scheme is said to be perfect if $\Delta = \mathcal{P} \setminus \Gamma$. Otherwise, the secret sharing scheme is said to be non-perfect.*

We can see that in a perfect secret sharing scheme, a subset of participants is either authorized or forbidden while in a non-perfect scheme, there is a subset which is neither authorized nor forbidden. In this thesis, unless otherwise stated, we assume that a secret sharing scheme is perfect. For a perfect secret sharing scheme, the privacy structure and the access structure determine each other.

When designing secret sharing schemes, it is important to make the size of the shares as small as possible. The following definition gives an important measure of efficiency of secret sharing schemes.

Definition 3.1.6. *The information rate of a secret sharing scheme Σ is defined as*

$$\rho(\Sigma) = \min\{H(\mathbf{S})/H(\mathbf{S}_i) : 1 \leq i \leq n\}.$$

The next theorem, proved in [31, 67], is an important result on the size of the shares in a perfect secret sharing scheme.

Theorem 3.1.7. *For any perfect secret sharing scheme, we have $H(\mathbf{S}_i) \geq H(\mathbf{S})$ for $1 \leq i \leq n$.*

Proof. Let $A \subseteq \mathcal{P}$ be a maximal forbidden/unauthorized set and let $P_i \notin A$. We have

$$\begin{aligned} H(\mathbf{S}_i) &\geq H(\mathbf{S}_i|\mathbf{S}_A) \\ &= H(\mathbf{S}|\mathbf{S}_A) + H(\mathbf{S}_i|\mathbf{SS}_A) - H(\mathbf{S}|\mathbf{S}_i\mathbf{S}_A) \\ &= H(\mathbf{S}) + H(\mathbf{S}_i|\mathbf{SS}_A) \\ &\geq H(\mathbf{S}). \end{aligned}$$

The first inequality is a basic property of the entropy. The first equality follows from the definition and property of the conditional mutual information $I(\mathbf{S}_i; \mathbf{S}|\mathbf{S}_A)$, that is,

$$I(\mathbf{S}_i; \mathbf{S}|\mathbf{S}_A) = H(\mathbf{S}_i|\mathbf{S}_A) - H(\mathbf{S}_i|\mathbf{SS}_A) \text{ and } I(\mathbf{S}_i; \mathbf{S}|\mathbf{S}_A) = I(\mathbf{S}; \mathbf{S}_i|\mathbf{S}_A).$$

The second equality is due to the fact that $H(\mathbf{S}|\mathbf{S}_A) = H(\mathbf{S})$ and $H(\mathbf{S}|\mathbf{S}_i\mathbf{S}_A) = 0$.

The last inequality follows from the nonnegativity of the entropy. $\square$

Theorem 3.1.7 implies that the information rate of any perfect secret sharing scheme is at most 1.

Definition 3.1.8. *A perfect secret sharing scheme Σ is said to be ideal if the information rate $\rho(\Sigma) = 1$.*

An alternative definition of the information rate is given by

$$\rho(\Sigma) = \min\{|\mathcal{S}|/|\mathcal{S}_i| : 1 \leq i \leq n\}.$$

In [70], it was proven that for any perfect secret sharing scheme, $|\mathcal{S}_i| \geq |\mathcal{S}|$ for $1 \leq i \leq n$. Hence, we can also say that a perfect scheme is ideal if the size of the

share for each participant is the same as the size of the secret. An access structure Γ is said to be *ideal* if it can be realized by an ideal secret sharing scheme.

Definition 3.1.9. *Given an access structure Γ on $\mathcal{P}$, we define the dual of Γ as*

$$\Gamma^\perp = \{A \subseteq \mathcal{P} : A^c \in \Gamma\}.$$

An access structure Γ is said to be self-dual if it is equal to its dual. Similarly, given a privacy structure Δ on $\mathcal{P}$, we define the dual of Δ as

$$\Delta^\perp = \{A \subseteq \mathcal{P} : A^c \in \Delta\}.$$

We can see that $(\Gamma^\perp)^\perp = \Gamma$ and $(\Delta^\perp)^\perp = \Delta$.

3.2 Some constructions

We survey some constructions of secret sharing schemes. Several constructions will also be discussed in other chapters.

3.2.1 Shamir threshold scheme

We present a construction of (t, n) -threshold scheme by A. Shamir [100]. The secrets and shares are elements of $\mathbb{F}_q$ where $q > n$. Suppose the secret is $s \in \mathbb{F}_q$. The dealer first selects n distinct non-zero elements of $\mathbb{F}_q$ which we will denote by $x_1, x_2, \dots, x_n$. These values are known by all participants. It then chooses $t-1$ elements $a_1, \dots, a_{t-1}$ from $\mathbb{F}_q$ independently at random and with uniform distribution. The next step is to construct the polynomial

$$f(x) = s + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}.$$

The share of P_i is given by $s_i = f(x_i)$ for $i = 1, \dots, n$.

Let $B = \{P_{i_1}, \dots, P_{i_t}\}$ be a t -subset of $\mathcal{P}$. The participants in B have t points of the polynomial $f(x)$. The Lagrange interpolation theorem guarantees that they can reconstruct the polynomial $f(x)$ using the formula

$$f(x) = \sum_{j=1}^t s_{i_j} \prod_{1 \leq k \leq t, k \neq j} \frac{x - x_{i_k}}{x_{i_j} - x_{i_k}}.$$

Hence, the participants in B can recover the secret by computing:

$$s = f(0) = \sum_{j=1}^t s_{i_j} \prod_{1 \leq k \leq t, k \neq j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}.$$

Now suppose a subset B of $t - 1$ participants tries to recover the secret. By the Lagrange interpolation theorem, for every $a \in \mathbb{F}_q$, the $t - 1$ points held by the participants in B together with the point $(0, a)$ determines a unique polynomial $f_a(x)$. This implies that the participants in B cannot rule out any value of the secret and that each possible value of the secret is equally probable.

3.2.2 KGH (n, n) -threshold scheme

Karnin, Greene and Hellman described in [67] a secret sharing scheme where the reconstruction of the secret requires all the shares. We call this scheme as the KGH (n, n) -threshold scheme. In this scheme, the set of possible secrets $\mathcal{S}$ is an abelian group and for $i = 1, \dots, n$, the set of possible shares $\mathcal{S}_i = \mathcal{S}$. To share a secret $s \in \mathcal{S}$, the dealer D selects independently at random $n - 1$ elements of $\mathcal{S}$, $y_1, \dots, y_{n-1}$ and then computes $y_n = s - \sum_{i=1}^{n-1} y_i$. For $1 \leq i \leq n$, D sends the share y_i to P_i .

We can see that the n participants can determine the secret. Any $n - 1$ participants can compute $s - y$ where y is the missing share. Since they do not know the value of y then they do not get any information about the secret.

3.2.3 Secret sharing schemes from cumulative arrays

We present a construction of secret sharing scheme for general access structures using cumulative arrays. The application of cumulative arrays to secret sharing schemes can be traced back to the work of Itoh, Saito and Nishizeki. In [61], they implicitly used the concept of cumulative arrays to prove that any monotone access structure can be realized by a perfect secret sharing scheme. Cumulative arrays were formally defined in [102]. The definitions and results in this subsection are taken from [74].

Definition 3.2.1. *Let Γ be an access structure on $\mathcal{P}$. Let $X = \{x_1, x_2, \dots, x_d\}$ and $\tau : \mathcal{P} \rightarrow 2^X$ be a mapping from $\mathcal{P}$ to the family of subsets of X . We call (τ, X) a cumulative array for Γ if the following condition is satisfied:*

$$\bigcup_{P \in I} \tau(P) = X \text{ if and only if } I \in \Gamma.$$

Let Γ be an access structure and let (τ, X) be a cumulative array for Γ with $X = \{x_1, x_2, \dots, x_d\}$. Our goal is to construct a secret sharing scheme realizing Γ . Given a secret $s \in \mathcal{S}$, the dealer first constructs a (d, d) -threshold scheme such as the KGH threshold scheme. Using this scheme, the dealer generates d shares $v_1, v_2, \dots, v_d$ for the secret. These shares are then distributed among the participants according to the cumulative array (τ, X) . For $1 \leq i \leq n$, the share of participant P_i is

$$s_i = \{v_j : x_j \in \tau(P_i)\}.$$

Each set $A \in \Gamma$ will have all the values $v_1, v_2, \dots, v_d$ and thus, A can reconstruct the secret. On the other hand, each set $B \notin \Gamma$, will not possess at least one v_i . It follows that B cannot obtain any information about the secret.

Definition 3.2.2. A cumulative array (τ, X) for Γ is called minimal if for any cumulative array (τ', X') for Γ , we have $|X'| \geq |X|$.

Consider the set $\mathcal{T} = \{T_1, \dots, T_l\}$ consisting of all maximal unauthorized sets with respect to Γ . Define the mapping τ as follows: for every $P_i \in \mathcal{P}$,

$$\tau(P_i) = \{T_j : P_i \notin T_j, 1 \leq j \leq l\}.$$

In other words, $\tau(P_i)$ is the collection of all maximal unauthorized sets not containing P_i . The pair $(\tau, \mathcal{T})$ is a minimal cumulative array for Γ [62].

In general, the secret sharing scheme obtained using cumulative arrays is very inefficient because the size of the shares is very large. For instance, suppose Γ is the (t, n) -threshold access structure and consider the minimal cumulative array $(\tau, \mathcal{T})$ for Γ . In the scheme obtained using $(\tau, \mathcal{T})$, the size of the share of each participant is $\binom{n-1}{t-1}$ times the size of the secret. Recall that in the Shamir threshold scheme, the size of the share of each participant is the same as the size of the secret. It is possible to improve this construction using generalized cumulative arrays. This will be tackled in Chapter 7.

3.3 Secret sharing schemes and matroids

We present in this section some of the important results on the relation between secret sharing schemes and matroids. We begin with the following definition from [11]:

Definition 3.3.1. Let Γ be an access structure on $\mathcal{P} = \{P_1, \dots, P_n\}$ and let $M = (Q, \mathcal{I})$ be a connected matroid. We say that the matroid M is appropriate for the

access structure Γ if $Q = \{0, 1, \dots, n\}$ and

$$\Gamma^- = \{C \setminus \{0\} : 0 \in C \text{ and } C \text{ is a circuit of } M\}.$$

The definition says that there is a 1-1 correspondence between the minimum access structure and the set of circuits of M containing 0. If a connected matroid is appropriate for an access structure, then that matroid is unique [78, 113].

One of the main open problems in the theory of secret sharing schemes is the characterization of ideal access structures. In [28], Brickell and Davenport gave a necessary (but not sufficient) condition for an ideal access structure.

Theorem 3.3.2. *If an access structure is ideal, then it has an appropriate matroid.*

This necessary condition is not sufficient as Seymour proved in [99] that the access structure that has the Vamos matroid as its appropriate matroid is not ideal. A sufficient (but not necessary) condition was also given in [28].

Theorem 3.3.3. *Let Γ be an access structure and suppose Γ has an appropriate matroid M . If M is representable over some finite field then Γ is an ideal access structure.*

This sufficient condition was proved to be not necessary by Simonis and Ashikhmin [103]. They showed that the access structure that has the non-Pappus matroid (which is not representable over any field) as its appropriate matroid is ideal.

3.4 Linear secret sharing schemes and monotone span programs

A *linear secret sharing scheme (LSSS)* is defined over a finite field $\mathbb{F}_q$. In such a scheme, the set of possible secrets and the set of possible shares of every participant are vector spaces over $\mathbb{F}_q$. Moreover, the secret is reconstructed using a linear function of the shares of authorized sets. Beimel [9] proved that we can replace the last statement by saying that each share is a linear function of the secret and some random field elements.

We present in this section the connection between linear secret sharing schemes and monotone span programs. A *monotone span program* is a linear algebraic model of computation introduced by Karchmer and Wigderson [66]. For simplicity, we only consider the case where the set of possible secrets is $\mathbb{F}_q$.

Definition 3.4.1. Let Γ be an access structure on $\mathcal{P} = \{P_1, \dots, P_n\}$. A *monotone span program (MSP)* $\mathcal{M}(\mathbb{F}_q, M, \epsilon, \psi)$ consists of

- i. an $l \times e$ matrix M over $\mathbb{F}_q$ where $e \leq l$;
- ii. a surjective function $\psi : \{1, \dots, l\} \rightarrow \{1, \dots, n\}$ labelling the rows of M ; and
- iii. a fixed vector $\epsilon = (1, 0, \dots, 0) \in \mathbb{F}_q^e$ called the target vector.

The size of $\mathcal{M}$ is defined as $\text{size}(\mathcal{M}) = l$.

We can think of ψ as a function assigning one or more rows to every participant in $\mathcal{P}$. In general, any nonzero vector can serve as a target vector for an MSP.

Given a subset $A \subseteq \mathcal{P}$ of participants, let M_A be the restriction of M to the rows assigned to the members of A . We say that $\mathcal{M}$ *accepts* A if $\epsilon \in \text{span}(M_A)$, where $\text{span}(M_A)$ is the linear span of the rows of M_A . If A is not accepted by $\mathcal{M}$ then we say that $\mathcal{M}$ *rejects* A . We say that an MSP $\mathcal{M}$ *computes the access structure* Γ if it accepts exactly those sets A such that $A \in \Gamma$.

We now prove two theorems to show the equivalence of monotone span programs and linear secret sharing schemes. Since linear secret sharing schemes and monotone span programs are equivalent then we will use the two terms interchangeably in this thesis. The first theorem was proved in [27, 28, 66].

Theorem 3.4.2. *Suppose there exists a monotone span program $\mathcal{M}$ over $\mathbb{F}_q$ that computes an access structure Γ . Then there is a linear secret sharing scheme over $\mathbb{F}_q$ realizing the access structure Γ .*

Proof. Let $\mathcal{M}(\mathbb{F}_q, M, \epsilon, \psi)$ be an MSP computing Γ . Suppose that the matrix M has e columns. We can obtain a linear secret sharing scheme realizing Γ from $\mathcal{M}$ as follows. To share a secret $s \in \mathbb{F}_q$, the dealer first chooses at random a vector $\rho \in \mathbb{F}_q^{e-1}$ then computes $M(s, \rho)^T$. The i th coordinate of $M(s, \rho)^T$ is given to participant $P_{\psi(i)}$. Since $\mathcal{M}$ computes Γ then for any $A \in \Gamma$, there exists a so-called recombination vector λ such that $\lambda M_A = \epsilon$. Using the recombination vector λ , the following relations holds:

$$\langle \lambda, (s, \rho) M_A^T \rangle = \langle \lambda M_A, (s, \rho) \rangle = \langle \epsilon, (s, \rho) \rangle = s$$

for any secret s and vector ρ .

Next, we want to show that any set $B \notin \Gamma$ cannot obtain any information about the secret. If $B \notin \Gamma$ then $\epsilon \notin \text{span}(M_B)$. Using linear algebra, this means that there

exists a vector $\mathbf{x} \in \mathbb{F}_q^e$ such that $M_B \mathbf{x}^T = \mathbf{0}^T$ and $\langle \boldsymbol{\varepsilon}, \mathbf{x} \rangle \neq 0$, i.e., the first coordinate of $\mathbf{x}$ is nonzero.

Suppose that the secret shared is $s \in \mathbb{F}_q$ and $\mathbf{w}_B$ is the vector of shares received by B , where $\mathbf{w}_B = M_B(s, \boldsymbol{\rho})^T$ for some $\boldsymbol{\rho} \in \mathbb{F}_q^{e-1}$. Let $\mathbf{y} = (s, \boldsymbol{\rho})$ and consider the set

$$Y = \{\mathbf{y}_a = \mathbf{y} + a\mathbf{x} : a \in \mathbb{F}_q\}.$$

For each $\mathbf{y}_a \in Y$, we have

$$M_B \mathbf{y}_a^T = M_B(\mathbf{y} + a\mathbf{x})^T = \mathbf{w}_B.$$

Moreover, since $\langle \boldsymbol{\varepsilon}, \mathbf{x} \rangle \neq 0$ then the set $\{\langle \boldsymbol{\varepsilon}, \mathbf{y}_a \rangle : \mathbf{y}_a \in Y\}$ consists of all the elements of $\mathbb{F}_q$. Hence, for each possible secret $s' \in \mathbb{F}_q$, there exists a vector $\mathbf{y}_a$ such that its first coordinate is s' and $M_B \mathbf{y}_a^T = \mathbf{w}_B$. Therefore, B cannot rule out any element of $\mathbb{F}_q$ as the secret. $\square$

The second theorem was proved by Beimel in [9].

Theorem 3.4.3. *Suppose there exists a linear secret sharing scheme over $\mathbb{F}_q$ realizing an access structure Γ . Then there is a monotone span program over $\mathbb{F}_q$ that computes the access structure Γ .*

Proof. Let Π be an LSSS realizing an access structure Γ with distribution mapping

$$\Sigma : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n,$$

where $\mathcal{S} = \mathbb{F}_q$, $\mathcal{S}_i$ are vector spaces over $\mathbb{F}_q$ for $1 \leq i \leq n$, and $\mathcal{R}$ is the set of random strings. Note that for each $(s, r) \in \mathcal{S} \times \mathcal{R}$, the distribution mapping Σ produces a vector of shares $\Sigma(s, r)$. Our goal is to construct an MSP $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ that computes Γ .

We begin with the construction of the matrix M . For $(s, r) \in \mathbb{F}_2 \times \mathcal{R}$, consider the vectors of shares $\Sigma(s, r)$. Then these vectors will be the columns of M . Thus, M will have $e = 2|\mathcal{R}|$ columns. Each column is indexed by a pair $(s, r) \in \mathbb{F}_2 \times \mathcal{R}$. The target vector is the binary vector $\boldsymbol{\varepsilon} = (s_1, s_2, \dots, s_e)$ where s_i is the secret corresponding to the i th column of M for $i = 1 \dots, e$.

Let l be the sum of the dimensions of $\mathcal{S}_i$ as vector spaces over $\mathbb{F}_q$, $i = 1, \dots, n$. Then the matrix M has l rows. By construction, the entries in a row of M are components of the shares of a participant. Thus, each row of M can be identified with a participant. This will be our definition of the labelling function ψ .

The next step is to prove that the MSP $\mathcal{M}$ computes the access structure Γ . We need to show that for $A \subseteq \mathcal{P}$, we have $A \in \Gamma$ if and only if $\boldsymbol{\varepsilon} \in \text{span}(M_A)$. Let $A \subseteq \mathcal{P}$ and suppose that $A \in \Gamma$. Then there exists a vector $\boldsymbol{\lambda}$ such that for every $(s, r) \in \mathbb{F}_2 \times \mathcal{R}$,

$$s = \langle \boldsymbol{\lambda}, \Sigma(s, r)|_A \rangle$$

where $\Sigma(s, r)|_A$ is the vector of shares of A . It follows that

$$\begin{aligned} \boldsymbol{\varepsilon} &= (s_1, s_2, \dots, s_e) \\ &= (\langle \boldsymbol{\lambda}, \Sigma(s_1, r_1)|_A \rangle, \dots, \langle \boldsymbol{\lambda}, \Sigma(s_e, r_e)|_A \rangle) \\ &= \langle \boldsymbol{\lambda}, M_A \rangle. \end{aligned}$$

Thus, $\boldsymbol{\varepsilon} \in \text{span}(M_A)$.

Suppose now that $\boldsymbol{\varepsilon} \in \text{span}(M_A)$. This implies that for every $(s, r) \in \mathbb{F}_2 \times \mathcal{R}$, the corresponding shares of A can always recover the secret s . Since the secret sharing scheme Π is perfect then A must be in the access structure Γ . $\square$

Given an MSP $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ of size l computing an access structure Γ , Gál [51] proved the existence of an MSP $\mathcal{M}^*(\mathbb{F}_q, M^*, \boldsymbol{\varepsilon}^*, \psi)$ computing the dual access structure $\Gamma^\perp$ and also of size l . This result was improved by Fehr [50] by showing that the MSP $\mathcal{M}^*$ can be efficiently computed from $\mathcal{M}$. The construction of $\mathcal{M}^*$ is as follows. Assume that M has l rows and e columns. Let $\mathbf{x}_0$ be a solution of the equation system $\mathbf{x}M = \boldsymbol{\varepsilon}$. Consider the subspace $\ker(M^T) = \{\mathbf{y} \in \mathbb{F}_q^l : \mathbf{y}M = \mathbf{0}\}$ and let $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{l-e}$ be a basis of $\ker(M^T)$. Then

$$M^* = [\mathbf{x}_0^T \ \mathbf{y}_1^T \ \mathbf{y}_2^T \ \cdots \ \mathbf{y}_{l-e}^T] \text{ and } \boldsymbol{\varepsilon}^* = (1, 0, 0, \dots, 0) \in \mathbb{F}_q^{l-e+1}.$$

The matrices M and M^* satisfy the relation $M^T M^* = \boldsymbol{\varepsilon} \cdot (\boldsymbol{\varepsilon}^*)^T$.

The idea behind monotone span programs already appeared in the so-called *vector space construction* by Brickell [27]. The vector space construction can be reformulated as an ideal monotone span program.

Suppose that we have an ideal access structure with an appropriate matroid that is representable over some finite field. The next two lemmas describe the relation between a matrix representation of the matroid and an MSP computing the access structure.

Lemma 3.4.4. *Let Γ be an ideal access structure for $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$. Suppose that the appropriate matroid $\mathcal{F}$ for Γ is $\mathbb{F}_q$ -representable and let $G = [\mathbf{g}_0 \ \mathbf{g}_1 \ \cdots \ \mathbf{g}_n]$ be a representation of $\mathcal{F}$ over $\mathbb{F}_q$, where $\mathbf{g}_i$ is the i th column of G . Let $M = [\mathbf{g}_1 \ \cdots \ \mathbf{g}_n]^T$, $\boldsymbol{\varepsilon} = \mathbf{g}_0^T$, and ψ a bijection of the set $\{1, 2, \dots, n\}$ to itself. Then $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ is an ideal MSP computing Γ .*

Lemma 3.4.5. *Let Γ be an ideal access structure for $\mathcal{P} = \{P_1, P_2, \dots, P_n\}$. Suppose that the appropriate matroid $\mathcal{F}$ for Γ is $\mathbb{F}_q$ -representable. Let $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ be an*

ideal MSP computing Γ . Then the matrix $G = [\epsilon^T \ M^T]$ is a representation of $\mathcal{F}$ over $\mathbb{F}_q$.

3.5 Secret sharing schemes from linear codes

We discuss two methods to construct secret sharing schemes using linear codes. The first method (Scheme I) was introduced by McEliece and Sarwate in [85] where they showed that the Shamir threshold scheme can be obtained using Reed-Solomon codes. The second method (Scheme II) was first studied by Karnin, Greene and Hellman [67] in the context of threshold schemes. It was investigated in the general case by Bertilsson and Ingemarsson [16], and by Massey [83, 84]. An important result by Massey is the connection between minimal authorized groups and minimal codewords.

3.5.1 Scheme I

The method used by McEliece and Sarwate was presented using a particular code, the Reed-Solomon code. Here, we describe the approach in a general way following the presentation in [96].

Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$ and consider a generator matrix G of $\mathcal{C}$. Suppose the secret to be shared is $s \in \mathbb{F}_q$. The dealer selects $k - 1$ elements $u_1, u_2, \dots, u_{k-1} \in \mathbb{F}_q$ independently and uniformly at random to form the information vector $(s, u_1, u_2, \dots, u_{k-1})$. Next, the dealer computes the corresponding codeword

$$(c_1, c_2, \dots, c_n) = (s, u_1, u_2, \dots, u_{k-1})G.$$

The share of participant P_i is the i th coordinate c_i . The resulting scheme is an ideal

linear secret sharing scheme.

We can reformulate the method described above in terms of monotone span programs. If we set $M = G^T$, $\epsilon = (1, 0, 0, \dots, 0) \in \mathbb{F}_q^k$ and ψ as a bijective map then we get an MSP $\mathcal{M}(\mathbb{F}_q, M, \epsilon, \psi)$. For $i = 1, 2, \dots, n$, let us denote the i th column of G by g_i^T . Then the next lemma follows from Theorem 3.4.2:

Lemma 3.5.1. *A set of shares $\{c_{i_1}, c_{i_2}, \dots, c_{i_m}\}$ uniquely determines the secret if and only if the vector ϵ^T can be expressed as a linear combination of the columns $g_{i_1}^T, g_{i_2}^T, \dots, g_{i_m}^T$.*

Consider now the Reed-Solomon code $\mathcal{C}$ (see Definition 2.1.22) given by

$$\mathcal{C} = \{(f(x_0), f(x_1), \dots, f(x_{n-1})) : f(x) \in \mathbb{F}_q[x], \deg(f) < k\}$$

where $x_0, x_1, \dots, x_{n-1}$ are distinct nonzero elements of $\mathbb{F}_q$. A generator matrix for $\mathcal{C}$ is

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_{n-1} \\ x_0^2 & x_1^2 & \cdots & x_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{k-1} & x_1^{k-1} & \cdots & x_{n-1}^{k-1} \end{bmatrix}.$$

We can see that the secret sharing scheme obtained from this code is the Shamir threshold scheme.

3.5.2 Scheme II

We now consider the method studied by Karnin, Greene and Hellman, Bertilsson and Ingemarsson, and by Massey. Let $\mathcal{C}$ be an $[n+1, k]$ linear code over $\mathbb{F}_q$ and let G be

a generator matrix of $\mathcal{C}$. Denote the i th column of G by $\mathbf{g}_i$, $i = 0, \dots, n$. Let $s \in \mathbb{F}_q$ be a secret. The dealer D randomly selects a vector $\mathbf{u} \in \mathbb{F}_q^k$ such that $\mathbf{u} \cdot \mathbf{g}_0 = s$. The dealer then computes the corresponding codeword

$$\mathbf{c} = (c_0, c_1, \dots, c_n) = \mathbf{u}G.$$

The share of P_i is c_i , for $i = 1, \dots, n$. Note that $c_0 = s$.

The resulting scheme is called the *secret sharing scheme based on $\mathcal{C}$* . The secret s can be determined by a set of shares $\{c_{i_1}, c_{i_2}, \dots, c_{i_r}\}$ if and only if $\mathbf{g}_0$ is a linear combination of $\mathbf{g}_{i_1}, \dots, \mathbf{g}_{i_r}$ where $1 \leq i_1 < \dots < i_r \leq n$. In particular, suppose that there exists constants $a_1, \dots, a_r$ such that $\mathbf{g}_0 = \sum_{j=1}^r a_j \mathbf{g}_{i_j}$. It follows that

$$s = \mathbf{u} \cdot \mathbf{g}_0 = \sum_{j=1}^r a_j (\mathbf{u} \cdot \mathbf{g}_{i_j}) = \sum_{j=1}^r a_j c_{i_j}.$$

In [83], it was shown that there is a relationship between the minimal authorized sets of the secret sharing scheme based on $\mathcal{C}$ and the minimal codewords of the dual code $\mathcal{C}^\perp$.

Theorem 3.5.2. *Let $\mathcal{C}$ be an $[n+1, k]$ -linear code over $\mathbb{F}_q$. In the secret sharing scheme based on $\mathcal{C}$, the set $\{P_{i_1}, \dots, P_{i_r}\} \subseteq \mathcal{P}$ such that $i_1 < \dots < i_r$ is a minimal authorized set if and only if there exists a minimal codeword $\mathbf{w} = (w_0, w_1, \dots, w_n) \in \mathcal{C}^\perp$ such that $\text{supp}(\mathbf{w}) = \{0, i_1, \dots, i_r\}$ and $w_0 = 1$.*

Proof. Let $A = \{P_{i_1}, \dots, P_{i_r}\} \subseteq \mathcal{P}$ be a minimal authorized set. Suppose $\{c_{i_1}, \dots, c_{i_r}\}$ is a set of shares of A . Then there exists nonzero constants $w_1, \dots, w_r$ such that $s = \sum_{j=1}^r w_j c_{i_j}$. This implies that the vector $\mathbf{w} = (1, -w_1, \dots, -w_n)$ whose support is $\{0, i_1, \dots, i_r\}$ is a codeword of $\mathcal{C}^\perp$. It is also a minimal codeword of $\mathcal{C}^\perp$ for otherwise,

if there is a codeword $\mathbf{v} \in \mathcal{C}^\perp$ such that $v_0 = 1$ and $\text{supp } \mathbf{v} \subset \text{supp } \mathbf{w}$ then this will contradict the minimality of A .

Suppose now that $\mathbf{w}$ is a minimal codeword of $\mathcal{C}^\perp$ with $w_0 = 1$. Let $\mathbf{c} \in \mathcal{C}$ be the codeword obtained in step 2 of the Massey construction above. Let $A = \{P_i \in \mathcal{P} : i \in \text{supp}(\mathbf{w}) \setminus \{0\}\}$. Since $\langle \mathbf{w}, \mathbf{c} \rangle = 0$ then the shares of A can determine the secret. If A is not a minimal authorized set then this will contradict the minimality of $\mathbf{w}$. $\square$

What is the relation between the SSS based on $\mathcal{C}$ and the SSS based on $\mathcal{C}^\perp$? The next lemma gives us an answer to this question.

Lemma 3.5.3. *Let $\Gamma_{\mathcal{C}}$ and $\Gamma_{\mathcal{C}^\perp}$ be the access structure of the SSS based on $\mathcal{C}$ and $\mathcal{C}^\perp$, respectively. Then $(\Gamma_{\mathcal{C}})^\perp = \Gamma_{\mathcal{C}^\perp}$.*

Proof. Let $A \subseteq \mathcal{P}$. By Theorem 3.5.2, $A \in \Gamma_{\mathcal{C}^\perp}$ if and only if there exists a codeword $\mathbf{c} \in \mathcal{C}$ such that $c_0 = 1$ and A contains the set $\text{supp}(\mathbf{c}) \setminus \{0\}$. For $i \in A^c$, we have $c_i = 0$. From linear algebra, this means that $A^c \notin \Gamma_{\mathcal{C}}$. Therefore, $A \in \Gamma_{\mathcal{C}^\perp}$ if and only if $A \in (\Gamma_{\mathcal{C}})^\perp$. $\square$

The following result was proved in [33]:

Theorem 3.5.4. *Let $\mathcal{C}$ be an $[n+1, k, d]$ -linear code over $\mathbb{F}_q$ with dual distance $d^\perp$. In the secret sharing scheme based on $\mathcal{C}$, every subset of at least $n-d+2$ participants is an authorized set while every subset of at most $d^\perp-2$ participants is a forbidden set.*

Proof. Theorem 3.5.2 implies that for every subset of participants A such that $|A| \leq d^\perp-2$ then $A \notin \Gamma_{\mathcal{C}}$. Similarly, for every subset of participants A such that $|A| \leq d-2$

then $A \notin \Gamma_{\mathcal{C}^\perp}$. Equivalently, for every subset of participants A such that $|A| \geq n-d+2$ then $A^c \notin \Gamma_{\mathcal{C}^\perp}$, that is, $A \in (\Gamma_{\mathcal{C}^\perp})^\perp$. It follows from Lemma 3.5.3 that $A \in \Gamma_{\mathcal{C}}$. $\square$

We now demonstrate the correspondence between MDS codes and ideal threshold schemes.

Theorem 3.5.5. *Let $\mathcal{C}$ be an $[n+1, k]$ MDS code over $\mathbb{F}_q$. Then the secret sharing scheme based on $\mathcal{C}$ is an ideal linear (k, n) -threshold scheme.*

Proof. It is clear that the secret sharing scheme based on $\mathcal{C}$ is ideal and linear. Since $\mathcal{C}$ is MDS then $d = n - k + 2$ and $d^\perp = k + 1$. If we apply Theorem 3.5.4 then every subset of at least k participants is in the access structure and every subset of at most $k - 1$ participants is not in the access structure. Therefore, the scheme is a (k, n) -threshold scheme. $\square$

Theorem 3.5.6. *Let Π be an ideal linear (k, n) -threshold scheme over $\mathbb{F}_q$. Then there exists an $[n+1, k]$ MDS code over $\mathbb{F}_q$.*

Proof. We can represent Π by an ideal MSP $\mathcal{M}(\mathbb{F}_q, M, \epsilon, \phi)$. We can assume that the columns of M are linearly independent. Consider the matrix $G = [\epsilon^T \mid M^T]$. We will show that G generates an $[n+1, k]$ MDS code over $\mathbb{F}_q$ by proving that every k columns of G are linearly independent and that the rank of G is equal to k .

Let $\mathbf{g}_{i_1}^T, \mathbf{g}_{i_2}^T, \dots, \mathbf{g}_{i_k}^T$ be k columns of G which does not include the first column ϵ^T . Since Π is a (k, n) -threshold scheme then ϵ^T is a linear combination of these k columns. Suppose that these k columns are linearly dependent. Then it follows that ϵ^T is a linear combination of less than k columns and we have a contradiction. Hence, every such k columns are linearly independent. This also implies that every

k columns which includes the first column $\boldsymbol{\varepsilon}^T$ must be linearly independent. Thus, every k columns of G are linearly independent and the rank of G is at least k .

Next we will prove that every $k + 1$ columns of G are linearly dependent. Let $\boldsymbol{g}_{i_1}^T, \boldsymbol{g}_{i_2}^T, \dots, \boldsymbol{g}_{i_k}^T, \boldsymbol{g}_{i_{k+1}}^T$ be $k + 1$ columns of G . If one of the columns is the first column $\boldsymbol{\varepsilon}^T$ then the $k + 1$ columns are linearly dependent since $\boldsymbol{\varepsilon}^T$ is a linear combination of the other k columns. Otherwise, there exist constants $a_1, a_2, \dots, a_k, b_2, b_3, \dots, b_{k+1}$ such that

$$\boldsymbol{\varepsilon} = \sum_{j=1}^k a_j \boldsymbol{g}_{i_j} = \sum_{j=2}^{k+1} b_j \boldsymbol{g}_{i_j}.$$

We then have

$$a_1 \boldsymbol{g}_{i_1} + \sum_{j=2}^k (a_j - b_j) \boldsymbol{g}_{i_j} - b_{k+1} \boldsymbol{g}_{i_{k+1}} = \mathbf{0}.$$

Consequently, the $k + 1$ columns $\boldsymbol{g}_{i_1}^T, \boldsymbol{g}_{i_2}^T, \dots, \boldsymbol{g}_{i_k}^T, \boldsymbol{g}_{i_{k+1}}^T$ are linearly dependent.

Since every $k + 1$ columns of G are linearly dependent then the rank of G is at most k . We already showed that the rank of G is at least k . Hence, the rank of G is equal to k . By Theorem 2.1.21, G is a generator matrix of an $[n + 1, k]$ MDS code over $\mathbb{F}_q$. □

By comparing Scheme I and Scheme II, we can see that in the former all the shares form a whole codeword while in the latter, the shares are just part of a codeword. Ding and Renvall also noted in [96] that the method of Scheme I may produce different schemes from the same code depending on the generator matrix used in the construction while the method of Scheme II does not depend on the generator matrix of the code.

The methods of construction of Scheme I and Scheme II are not the same but there is a relation between the two as noted by Van dijk [110]. Suppose we apply

the method of Scheme I to a linear code $\mathcal{C}$ with generator matrix G . We obtain the same scheme by applying the method of Scheme II to the linear code $\mathcal{C}'$ generated by the matrix $G' = [\boldsymbol{\varepsilon}^T \mid G]$ where $\boldsymbol{\varepsilon} = (1, 0, 0, \dots, 0)$. If we start with a secret sharing scheme based on a linear code $\mathcal{C}$ then we can obtain the same scheme by applying the method of Scheme I to the code obtained by puncturing $\mathcal{C}$ on the first coordinate position.

3.6 Multiplicative linear secret sharing schemes

We discuss in this section the notion of *multiplicativity* and *strong multiplicativity* in linear secret sharing schemes. These two concepts are properties of LSSS introduced by Cramer, Damgård and U. Maurer in [36]. Multiplicative LSSS (MLSSS) and strongly multiplicative LSSS (SMLSSS) are used as building blocks in protocols for secure multiparty computation.

Definition 3.6.1. *Let $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ be an ideal LSSS realizing an access structure Γ . Then $\mathcal{M}$ is said to be multiplicative if there exists a vector $\mathbf{r}$ such that for any two secrets s, s' and respective share vectors*

$$(x_1, \dots, x_n) = M(s, \boldsymbol{\rho})^T, (x'_1, \dots, x'_n) = M(s', \boldsymbol{\rho}')^T,$$

where $\boldsymbol{\rho}, \boldsymbol{\rho}'$ are random vectors, it holds that

$$ss' = \langle \mathbf{r}, (x_1 x'_1, \dots, x_n x'_n) \rangle.$$

We say that $\mathcal{M}$ is strongly multiplicative if for any subset $A \notin \Gamma$, the restriction $\mathcal{M}_{A^c}$ of $\mathcal{M}$ to participants in A^c is multiplicative.

The vector $\mathbf{r}$ in the definition is called the *recombination vector*. The definition can be extended to LSSS that are not ideal and even to non-perfect schemes. The main idea is that in a multiplicative LSSS, given two secrets s and s' , each participant P_i can compute a value c_i using only its own shares of the two secrets in such a way that the product ss' is a linear combination of $c_1, c_2, \dots, c_n$. In a strongly multiplicative LSSS, we add the condition that for any unauthorized set A , the product ss' can be obtained using only the values c_i from participants not in A .

Given a linear secret sharing scheme, it is natural to ask how to know whether it is multiplicative or not. The answer depends on the corresponding access structure.

Definition 3.6.2. *Let d be an integer such that $d \geq 2$. An access structure Γ is called Q_d if for every d sets $A_1, A_2, \dots, A_d$ in Γ^c , we have $A_1 \cup A_2 \cup \dots \cup A_d \subset \mathcal{P}$.*

In other words, an access structure Γ is Q_d if the set of participants $\mathcal{P}$ cannot be covered by d unauthorized subsets. We say that a privacy structure $\Delta = \Gamma^c$ is Q_d if the corresponding access structure is Q_d .

The most interesting cases are when $d = 2, 3$. We note that an access structure Γ is Q_2 if and only if $\Gamma^\perp \subseteq \Gamma$ [36]. As an example, the (t, n) -threshold access structure is Q_2 if $n \geq 2t - 1$ and is Q_3 if $n \geq 3t - 2$. The next result was proved in [36].

Theorem 3.6.3. *Let $\mathbb{F}_q$ be a finite field and let Γ be an access structure. Then there exists an MLSSS (resp. SMLSSS) over $\mathbb{F}_q$ realizing Γ if and only if Γ is Q_2 (resp. Q_3).*

Example 3.6.4. *Consider the Shamir (t, n) -threshold scheme over $\mathbb{F}_q$. Let s, s' be two secrets with share vectors $(s_1, s_2, \dots, s_n), (s'_1, s'_2, \dots, s'_n)$ respectively. We know that these secrets and shares are generated by some polynomials of degree at most*

$t - 1$, say $f(x)$ and $g(x)$, such that $f(0) = s, f(x_i) = s_i, g(0) = s', g(x_i) = s'_i$ for $1 \leq i \leq n$.

Since the degree of the polynomial fg is at most $2t - 2$ then by the Lagrange interpolation theorem, it is enough to have at least $2t - 1$ points to reconstruct the polynomial fg . Thus, if $n \geq 2t - 1$ then the Shamir (t, n) -threshold scheme is multiplicative. Furthermore, if $n \geq 3t - 2$ then it is strongly multiplicative since $n - (t - 1) \geq 2t - 1$.

Example 3.6.5. Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathbb{F}_q$ that is self-dual. Then the SSS based on $\mathcal{C}$ is multiplicative [33]. Indeed, given two codewords $(c_0, c_1, \dots, c_n)$ and $(c'_0, c'_1, \dots, c'_n)$ of $\mathcal{C}$, we have

$$c_0 c'_0 = - \sum_{i=1}^n c_i c'_i.$$

Given a LSSS realizing a Q_2 access structure, it can be transformed into a multiplicative scheme for the same access structure [36]. In general, the transformation involves doubling the size of the shares in the original scheme. This is not the case for strong multiplicativity, as it is not known how to efficiently construct strongly multiplicative schemes from LSSS realizing Q_3 access structures.

It is an open problem to determine the LSSS which do not require share expansion to obtain the multiplicative property. A particular case of this problem is the characterization of access structures of ideal multiplicative LSSS. The (t, n) -threshold access structures with $n \geq 2t - 1$ can be realized by an ideal multiplicative LSSS (e.g. Shamir scheme). Given a self-dual access structure which can be realized by an ideal LSSS, the question of whether it admits an ideal multiplicative scheme was studied in [37, 91]. Self-dual access structures are the minimally Q_2 access structures. This means that for every subset Γ' of a self-dual access structure Γ then Γ' is not Q_2 . It

was shown in [37], using the relation between matroids and codes, that the answer is affirmative for such access structures which are bipartite. It was proven in [91] that such access structures with at most seven participants admit an ideal multiplicative LSSS.

We have already seen that LSSS based on self-dual codes and some Shamir threshold schemes are ideal MLSSS. Ideal multiplicative schemes can also be obtained from algebraic geometric LSSS [32]. A construction for hierarchical access structures can be found in [68]. The authors in [73] and in [87] presented constructions of ideal multiplicative schemes of some access structures based on graphs.

We end this section by illustrating how a multiplicative LSSS is used in constructing secure multiparty computation protocols (MPC). Our main reference for this topic is [36]. In a secure MPC, there are n participants who want to jointly compute a certain function. Each participant has its own input to the function. The computation must be performed in such a way that the output is correct and the input of each participant remains private even if some of the participants behave maliciously. The set of all possible groups of malicious participants is called an *adversary structure*.

It is generally assumed that the function (to be computed) is represented by an arithmetic circuit over some finite field consisting of addition and multiplication gates. It follows that it is sufficient to construct protocols for secure addition and secure multiplication.

Let $\mathcal{M}(\mathbb{F}_q, M, \varepsilon, \psi)$ be a multiplicative MSP computing an access structure Γ . To simplify the exposition, we consider the case where $\mathcal{M}$ is ideal and the adversary structure is $\Gamma^{\complement}$. We also assume that participants belonging to a group in the adversary structure try to learn the inputs of the others but still follow the protocol (this

is called *passive adversary* in the literature).

The secret sharing scheme $\mathcal{M}$ is used as follows. At the start, each participant shares its input using $\mathcal{M}$. The arithmetic circuit is then processed gate by gate. The intermediate inputs and outputs must also be secret-shared using $\mathcal{M}$. Suppose the inputs to a gate is a and b , with respective shares

$$(a_1, a_2, \dots, a_n) = M(a, \boldsymbol{\rho}_a)^T \text{ and } (b_1, b_2, \dots, b_n) = M(b, \boldsymbol{\rho}_b)^T.$$

To perform secure addition, each participant P_i computes $a_i + b_i$ for $1 \leq i \leq n$.

By the linearity of $\mathcal{M}$, we have

$$(a_1 + b_1, a_2 + b_2, \dots, a_n + b_n) = M(a, \boldsymbol{\rho}_a)^T + M(b, \boldsymbol{\rho}_b)^T = M(a + b, \boldsymbol{\rho}_a + \boldsymbol{\rho}_b)^T.$$

Hence, $a_1 + b_1, a_2 + b_2, \dots, a_n + b_n$ are the shares for $a + b$.

To perform secure multiplication, first we note that since $\mathcal{M}$ is multiplicative then there exists a recombination vector $\mathbf{r} = (r_1, r_2, \dots, r_n)$ such that

$$c = ab = \sum_{i=1}^n r_i(a_i b_i).$$

For $i = 1, 2, \dots, n$, participant P_i computes $c'_i = a_i b_i$ and shares c'_i using $\mathcal{M}$. If $(c_{i1}, c_{i2}, \dots, c_{in})$ are the shares generated for c'_i then c_{ij} is sent to P_j . For $j = 1, 2, \dots, n$, participant P_j computes

$$c_j = \sum_{i=1}^n r_i c_{ij}.$$

It can be shown using the same argument as in the case of secure addition that $c_1, c_2, \dots, c_n$ are the shares for $c = ab$.

3.7 Non-perfect secret sharing schemes

We recall that a secret sharing scheme is non-perfect if $\Gamma \cup \Delta \neq \mathcal{P}$. This means that there exists a subset A of participants that has partial information about the secret, i.e. $0 < H(\mathbf{S}|\mathbf{S}_A) < H(\mathbf{S})$. The main advantage of non-perfect schemes is that the size of the share can be smaller than the size of the secret which can be useful in situations where the secret is very large.

A well-known example of a non-perfect scheme is the ramp scheme. This scheme was introduced by Blakley and Meadows [21]. The following definition is taken from [63].

Definition 3.7.1. *Let n, t_1, t_2 be integers such that $0 \leq t_1 < t_2 \leq n$. A (t_1, t_2, n) -ramp scheme is a secret sharing scheme realizing (Γ, Δ) such that*

$$\Gamma = \{A \subseteq \mathcal{P} : |A| \geq t_2\} \text{ and } \Delta = \{B \subseteq \mathcal{P} : |B| \leq t_1\}.$$

A (t_1, t_2, n) -ramp scheme is said to be linear if for every $A \subseteq \mathcal{P}$ with $|A| = t$, where $t_1 \leq t \leq t_2$, we have

$$H(\mathbf{S}|\mathbf{A}) = \frac{t_2 - t}{t_2 - t_1} H(\mathbf{S}).$$

The Shamir threshold scheme can be extended to a ramp scheme.

Example 3.7.2. *The secrets and shares are elements of $\mathbb{F}_q$ where $q \geq n + t_2 - t_1$. Suppose the secrets are $s_1, s_2, \dots, s_{t_2-t_1} \in \mathbb{F}_q$. The dealer first selects $n + t_2 - t_1$ distinct elements of $\mathbb{F}_q$ which we will denote by $c_1, \dots, c_{t_2-t_1}, x_1, \dots, x_n$. The values $x_1, \dots, x_n$ are known by all participants. It then chooses t_2 elements $a_0, a_1, \dots, a_{t_2-1}$ from $\mathbb{F}_q$ independently at random and with uniform distribution. The next step is to*

construct the polynomial

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t_2-t_1}x^{t_2-t_1}$$

such that $f(c_i) = s_i, i = 1, \dots, t_2 - t_1$. The share of P_i is given by $f(x_i)$ for $i = 1, \dots, n$. The information rate of the scheme is equal to $t_2 - t_1 > 1$.

We can see that a subset $B \subseteq \mathcal{P}$ of at most t_1 participants cannot obtain any information about the secret. On the other hand, a subset B of at least t_2 participants can determine all the secrets. A subset B of t participants where $t_1 < t < t_2$ can have partial information about the secret.

The theorem below gives bounds on the size of shares in ramp schemes.

Theorem 3.7.3. *Let Π be a (t_1, t_2, n) -ramp scheme. For any subset $A \subseteq \mathcal{P}$ with $|A| = t_2 - t_1$ we have $H(\mathbf{A}) \geq H(\mathbf{S})$. Moreover, if Π is linear then for any subset $B \subseteq \mathcal{P}$, we have $H(\mathbf{B}) \geq |B|H(\mathbf{S})/(t_2 - t_1)$.*

The first result is taken from [23]. It implies that the average size of the share of a participant is at least $H(\mathbf{S})/(t_2 - t_1)$. The second result first appeared in [30] but with an incomplete proof. The complete proof was given in [63]. The second result says that in linear ramp schemes, the size of the share of a participant is at least $H(\mathbf{S})/(t_2 - t_1)$ (this improves the first result). The scheme in Example 3.7.2 is linear and we can see that the size of the share of each participant is equal to $H(\mathbf{S})/(t_2 - t_1)$. In this sense, the scheme is said to be optimal.

For general non-perfect secret sharing schemes, Ogata and Kurosawa proved in [88] that there exists a non-perfect SSS realizing (Γ, Δ) if and only if both Γ and Δ^c are monotone increasing. They also proved the following lower bounds on the size of the shares.

Lemma 3.7.4. *In any non-perfect SSS realizing (Γ, Δ) , we have*

$$\sum_{P_i \in A \setminus B} H(\mathbf{S}_i) \geq H(\mathbf{S})$$

for any $A \in \Gamma$ and $B \in \Delta$ such that $B \subset A$.

Theorem 3.7.5. *In any non-perfect SSS realizing (Γ, Δ) , we have*

$$\max_{1 \leq i \leq n} \{\log |\mathcal{S}_i|\} \geq H(\mathbf{S}) / \min \{|A \setminus B|\}$$

where the minimum is taken over all $A \in \Gamma, B \in \Delta$.

Proof. Let $A \in \Gamma, B \in \Delta$. If $B \subset A$ then by the preceding lemma and by the property of the entropy

$$H(\mathbf{S}) \leq \sum_{P_i \in A \setminus B} H(\mathbf{S}_i) \leq (|A \setminus B|) \max_{1 \leq i \leq n} \{\log |\mathcal{S}_i|\}.$$

Assume now that $B \not\subset A$. Consider the set $C = A \cup B$. Since Γ is monotone increasing then $C \in \Gamma$. Since $B \subset C$ then we have

$$H(\mathbf{S}) \leq (|C \setminus B|) \max_{1 \leq i \leq n} \{\log |\mathcal{S}_i|\}.$$

Note that $|C \setminus B| = |A \setminus B|$. Hence,

$$H(\mathbf{S}) \leq (|A \setminus B|) \max_{1 \leq i \leq n} \{\log |\mathcal{S}_i|\}.$$

□

3.8 Summary

We have given an overview of the theory of secret sharing schemes. We used the definition in terms of the entropy function in information theory.

We also described three classical constructions of secret sharing schemes. The first two are for threshold schemes: the Shamir (t, n) -threshold scheme and the KGH (n, n) -scheme. The third method is the construction of secret sharing schemes for general access structures using cumulative arrays. We presented the connection between ideal access structures and matroids. We also talked about the equivalence between linear SSS and monotone span programs.

We discussed the two main constructions of SSS from linear codes. The first one was introduced by McEliece and Sarwate while the second was considered by Karnin, Greene and Hellman (threshold case), by Bertilsson and Ingemarsson and by Massey (general case).

The concept of multiplicative linear SSS and its significance in secure multiparty computation was also discussed. The last section contains a discussion of non-perfect SSS.

Chapter 4

Secret sharing schemes based on graphical codes

This chapter deals with secret sharing schemes based on codes from complete graphs. We describe the corresponding access structures and look into the multiplicativity of the schemes. Multiplicative LSSS are used in designing protocols for secure multiparty computation. It is an open problem to characterize the access structures that can be realized by ideal multiplicative LSSS.

In Section 4.1, we analyze the access structure of LSSS based on the cut-set and cycle codes and prove that these schemes cannot be multiplicative. In Section 4.2, we study the access structure based on odd cycles. This is the access structure of the scheme based on the dual of the extended cycle code. We prove that this access structure does not admit any ideal multiplicative LSSS. This can be seen as a contribution to the problem of determining the access structures that admit ideal multiplicative schemes. In Section 4.3, we consider an extension of the access structure

based on odd cycles and show that it can be realized by ideal multiplicative LSSS. This access structure corresponds to the scheme based on the dual of an augmented extended cycle code. This chapter is a joint work with Gao Ying [52].

4.1 Secret sharing schemes based on cut-set and cycle codes

Many researchers focused on studying the secret sharing schemes for graph-based access structure with the minimal access structure being the collection of the pairs of participants corresponding to edges, i.e., associate each participant with a vertex of the underlying graph, and any two participants can recover the secret if there is an edge connecting them. See [106] and its references.

Another way to construct an access structure based on a graph G is first to associate each participant with an edge and then consider certain subsets of the edge set E . For example, Karchmer and Wigderson [66] considered access structures based on the connectivity of two designated vertices. For complete graphs, Liu *et al.* [73] considered the set of all spanning trees while Beimel [9] considered the set of all odd cycles.

We consider the secret sharing schemes based on cut-set codes and cycle codes associated with K_m . Note that the coordinate positions are indexed by the edges of the graph. The dealer D is associated with the edge corresponding to the first coordinate position and each of the $n - 1$ participants is associated with the rest of the edges/coordinate positions. First, we describe the access structure of the secret

sharing scheme based on the cut-set code $\mathcal{C}(K_m)$.

Proposition 4.1.1. *Let v_s and v_t be the vertices of the edge corresponding to the first coordinate position. In the secret sharing scheme based on $\mathcal{C}(K_m)$, a set of shares $\{c_{i_1}, c_{i_2}, \dots, c_{i_r}\}$ can determine the secret if and only if the corresponding set of edges contains a path from v_s to v_t .*

Proof. It follows from Lemma 3.5.2 that a set of shares $\{c_{i_1}, c_{i_2}, \dots, c_{i_r}\}$ can determine the secret if and only if there exists a codeword $\mathbf{z} = (z_0, \dots, z_{n-1}) \in \mathcal{C}^\perp(K_m)$ such that $z_0 = 1$ and $\text{supp}(\mathbf{z}) \subseteq \{0, i_1, \dots, i_r\}$. Since the dual code of $\mathcal{C}(K_m)$ is the cycle code $\mathcal{C}^*(K_m)$ then the support of $\mathbf{z}$ corresponds to a cycle or a union of edge-disjoint cycles. Therefore, a set of shares $\{c_{i_1}, c_{i_2}, \dots, c_{i_r}\}$ can determine the secret if and only if the corresponding set of edges contains a path from v_s to v_t . $\square$

The access structure realized by the secret sharing scheme based on $\mathcal{C}(K_m)$ is called the undirected s - t connectivity access structure Γ_{ustcon} on K_m . In [10], a MSP computing Γ_{ustcon} was presented. By using Lemma 3.4.5, we can get a generator matrix of $\mathcal{C}(K_m)$ from the MSP.

Now, we consider the access structure of the secret sharing scheme based on the cycle code $\mathcal{C}^*(K_m)$.

Proposition 4.1.2. *Every nonzero codeword of the cut-set code $\mathcal{C}(K_m)$ is minimal.*

Proof. First, we recall that a generator matrix for $\mathcal{C}(K_m)$ is the reduced incidence matrix H_m . By looking at the generator matrix H_m , we can see that for any two nonzero codewords $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}(K_m)$, $\text{supp}(\mathbf{c}_1) \cap \text{supp}(\mathbf{c}_2)$ is non-empty. The proposition now follows from Lemma 2.1 (5) in [5]. $\square$

Corollary 4.1.3. *In the secret sharing scheme based on $\mathcal{C}^*(K_m)$, there are 2^{m-2} minimal authorized sets and each participant P_i belongs to 2^{m-3} out of 2^{m-2} minimal authorized sets.*

Proof. The result follows from the preceding proposition and Proposition 2 in [44]. $\square$

If we want to know whether we can get multiplicative linear secret sharing schemes based on the linear codes above, then the following propositions say that it is not the case.

Proposition 4.1.4. *Consider the access structure Γ_{ustcon} based on the complete graph K_m . Then Γ_{ustcon} is not Q_2 .*

Proof. First, let the edge v_1v_m correspond to the dealer. Hence, to prove the proposition, we need to find two unauthorized sets of edges whose union cover the set $E \setminus \{v_1v_m\}$. Let A_1 be the subset of $E \setminus \{v_1v_m\}$ containing the edges incident with v_1 but not with v_m . Then $A_1 \notin \Gamma_{ustcon}$ since it does not contain a path from v_1 to v_m . Let $A_2 = E \setminus (\{v_1v_m\} \cup A_1)$. Then A_2 is also not an authorized set. Now, $A_1 \cup A_2$ is the whole participant set $E \setminus \{v_1v_m\}$. Therefore, Γ_{ustcon} is not Q_2 . $\square$

Proposition 4.1.5. *The access structure realized by the secret sharing scheme based on the cycle code $\mathcal{C}^*(K_m)$ is not Q_2 for $m \geq 4$.*

Proof. Let Γ be the access structure realized by the secret sharing scheme based on $\mathcal{C}^*(K_m)$ and let v_1v_m be the edge associated with the dealer. First, we note that $A \subseteq \mathcal{P}$ is in the minimal access structure Γ^- if and only if $A \cup \{v_1v_m\}$ is a bond (minimal cut-set) [11].

The secret sharing scheme based on $\mathcal{C}^*(K_3)$ is a $(1, 2)$ -threshold scheme. Hence, the corresponding access structure is Q_2 and we know that the corresponding Shamir scheme is ideal and multiplicative.

Now let $m \geq 4$. Suppose, for a contradiction, that Γ is Q_2 . Then it follows that the dual access structure $\Gamma^\perp \subseteq \Gamma$ [36]. Now, $\Gamma^\perp$ is the access structure realized by the secret sharing scheme based on $\mathcal{C}(K_m)$. Let $A \in \Gamma^\perp$ such that $A \cup \{v_1 v_m\}$ is a 3-cycle. Then $A \notin \Gamma$, contradicting the assumption that Γ is Q_2 . $\square$

4.2 Access structure based on odd cycles

We revisit a class of access structures which appears in [9] and where the minimal authorized sets correspond to odd cycles. These access structures are related to the dual of the extended cycle code and can be realized by multiplicative linear secret sharing schemes (unlike in the case of the access structures in the previous section). One drawback is that we will prove that there is no ideal multiplicative linear secret sharing scheme over $\mathbb{F}_q$ that realize them. The negative result, though, is a contribution to the characterization of access structures which admit ideal multiplicative LSSS.

4.2.1 Definition and some properties

Definition 4.2.1. *Let K_m be the complete graph with m vertices and let n be the number of edges. For $i = 1, \dots, n$, every participant P_i is assigned to an edge. Define*

$$\Gamma_{oc} = \{A \subseteq \mathcal{P} : K_m[A] \text{ contains an odd cycle}\}$$

Then Γ_{oc} is an access structure.

It follows from the definition that the privacy structure is given by

$$\Delta_{oc} = \{B \subseteq \mathcal{P} : K_m[B] \text{ does not contain an odd cycle}\}.$$

We can see that Γ_{oc} is monotone increasing while Δ_{oc} is monotone decreasing. Using Lemma 2.2.2, we can also define Γ_{oc} and Δ_{oc} in terms of bipartite graphs.

Proposition 4.2.2. *The access structure Γ_{oc} is Q_d if and only if $m \geq 2^d + 1$.*

Proof. The complete graph K_m can be expressed as the union of k bipartite graphs if and only if $m \leq 2^k$ (cf. [114]). The proposition now follows. $\square$

It follows from the proposition that for $m \geq 5$, the access structure Γ_{oc} on K_m can be realized by a multiplicative LSSS.

4.2.2 Appropriate matroid

Let $G(V, E)$ be a connected undirected graph. By attaching a sign to each edge of G , we obtain a *signed graph* G_s . The sign of a cycle is the product of the signs of its edges.

We consider two matroids defined on the edge set of a signed graph G_s . The *lift matroid* $L(G_s)$ is the matroid wherein a circuit is a positive cycle of G_s or the disjoint union of two negative cycles of G_s that meet in at most one vertex. The *complete lift matroid* $L_0(G_s)$ is an extension of $L(G_s)$ obtained by adding an extra point e_0 (which acts like a negative loop) to $E(G_s)$ (the point set of $L(G_s)$). A circuit of $L_0(G_s)$ is a circuit of $L(G_s)$ or the union of e_0 and a negative cycle.

Given $M(G)$ and the incidence matrix H of G , we can obtain the binary representation of the complete lift matroid $L_0(G_s)$ [53]. First, we define the row incidence vector d of the signs of the edges. For $i = 1, \dots, n$, let $d_i = 1$ if the corresponding edge is negative, and 0 otherwise. The binary representation of $L_0(G_s)$ is given by

$$H' = \left[\begin{array}{c|c} 1 & d \\ \hline 0 & \\ \vdots & H \\ 0 & \end{array} \right].$$

Note that we can replace H by the reduced incidence matrix H_m .

Example 4.2.3. Let G_s be the complete graph K_m with all edges negative, denoted by $-K_m$. Suppose we add a negative loop e_0 at the vertex v_m . A circuit of $L_0(-K_m)$ is the disjoint union of two odd cycles that meet in at most one vertex or the union of e_0 and an odd cycle. The binary representation of $L_0(-K_m)$ is given by H' above, with the vector d equal to the all-one vector.

Based on the preceding example, we can see that $L_0(-K_m)$ is the appropriate matroid for the access structure Γ_{oc} on K_m . Consider the extended cycle code $\widehat{\mathcal{C}}^*(K_m)$. The binary representation H' with the matrix H replaced by H_m is a parity check matrix of $\widehat{\mathcal{C}}^*(K_m)$. Hence, Γ_{oc} is the access structure realized by the secret sharing scheme based on the dual of $\widehat{\mathcal{C}}^*(K_m)$.

Next, we want to show that for $m \geq 4$, $L_0(-K_m)$ is only representable over a field with characteristic 2. We will use the following proposition taken from [53] (Note: For the meaning of the concept of minor and F_7^* , the reader may refer to [90].):

Proposition 4.2.4. *Let G_s be a signed graph. Then $L_0(G_s)$ has an F_7^* minor using e_0 if and only if G_s contains $-K_4$.*

Corollary 4.2.5. *For $m \geq 4$, $L_0(-K_m)$ is only representable over a field with characteristic 2.*

Proof. From the preceding proposition, $L_0(-K_m)$ has an F_7^* minor using e_0 . If a matroid is representable over a field $\mathbb{F}_q$ then its minor is also representable over the same field. It is well-known that F_7^* can only be represented over a field with characteristic 2. $\square$

Corollary 4.2.6. *The access structure Γ_{oc} defined on K_m , $m \geq 4$, can only be realized by an ideal LSSS over a field of characteristic 2.*

4.2.3 LSSS realizing the access structure based on odd cycles

By Lemma 3.4.4 and using the binary representation of the complete lift matroid $L_0(-K_m)$, we get an ideal MSP realizing Γ_{oc} over $\mathbb{F}_2$. We show that the same MSP computes Γ_{oc} over any field of characteristic 2.

Proposition 4.2.7. *Let K_m be the complete graph on m vertices and $\mathbb{F}_q$ be a field of characteristic 2. Let M be an $(n \times m)$ matrix over $\mathbb{F}_q$ defined as*

$$M = \begin{bmatrix} \mathbf{1}^T & H_m^T \end{bmatrix},$$

where $\mathbf{1}$ is the all-one row vector and H_m is a reduced incidence matrix of K_m .

Then $\mathcal{M}(\mathbb{F}_q, M, \varepsilon, \psi)$ is an ideal MSP computing the access structure Γ_{oc} where $\varepsilon = (1, 0, 0, \dots, 0)$ and ψ is a bijection.

Proof. Assume that $H_m^T = [\mathbf{h}_1^T \ \mathbf{h}_2^T \ \cdots \ \mathbf{h}_{m-1}^T]$ where $\mathbf{h}_i$ is the incidence vector of the vertex v_i . Given a secret s , the shares are generated by choosing $m-1$ random values $r_1, \dots, r_{m-1}$ and then computing $M(s, r_1, \dots, r_{m-1})^T$. If a participant corresponds to an edge (v_k, v_l) where k, l are both not equal to m then its share is given by $x_{kl} = s + r_k + r_l$. If a participant corresponds to an edge (v_k, v_m) then its share is given by $x_{km} = s + r_k$.

Let $A \subseteq \mathcal{P}$. Suppose A corresponds to an odd cycle. Then the first column of M_A have an odd number of 1's while the other columns have an even number of 0's. Hence, taking the sum of the rows of M_A gives us the target vector $(1, 0, \dots, 0)$. Therefore, $A \in \Gamma_{oc}$.

Suppose A does not contain an odd cycle, or equivalently, A is bipartite. For a secret s , let $\{x_{kl} : (v_k, v_l) \in A\}$ be the set of shares of participants in A . We want to show that the number of random values to generate the shares $\{x_{kl} : (v_k, v_l) \in A\}$ given the secret s is equal to the number of random values that generate the same set of shares given the secret s' where $s \neq s'$.

Since A is bipartite then we can partition the set of vertices corresponding to A into two sets V_1, V_2 such that for any edge in A , one of its endpoints is in V_1 while the other is in V_2 . Choose one V_i which does not contain v_m . Note that it is possible that v_m is not contained in both V_1 and V_2 . In this case, simply choose any set. Let V' be the selected set. For $i = 1, \dots, m-1$, define $r'_i = r_i + (s - s')$ if $v_i \in V'$ and $r'_i = r_i$ otherwise. We will show that $r'_1, \dots, r'_{m-1}$ generate the shares $\{x_{kl} : (v_k, v_l) \in A\}$.

Let (v_k, v_l) be an edge in A where k, l are both not equal to m . Then only one of

the endpoints, say v_k , is in V' . It follows that

$$x'_{kl} = s' + r'_k + r'_l = s' + r_k + (s - s') + r_l = s + r_k + r_l = x_{kl}.$$

Suppose now that (v_k, v_m) is an edge in A . Then we have

$$x'_{km} = s' + r'_k = s' + r_k + (s - s') = s + r_k = x_{km}.$$

By construction, we can see that the MSP is ideal. □

4.2.4 Multiplicativity

We now consider the question: Can the access structure Γ_{oc} be realized by an ideal multiplicative LSSS? If $m < 5$ then we know from Proposition 4.2.2 that the access structure Γ_{oc} is not Q_2 . Hence, we assume that $m \geq 5$.

We will use the following alternative way to decide whether or not an ideal MSP is multiplicative [73]. First, we define an operation $*$ on a matrix.

Definition 4.2.8. Let P be an $(n \times l)$ matrix over $\mathbb{F}_q$ and suppose that the i -th column of P is $\mathbf{p}_i = (p_{1i}, \dots, p_{ni})^T$. Let P^* be the matrix constituted by all the column vectors

$$\mathbf{p}_i * \mathbf{p}_j = (p_{1i}p_{1j}, \dots, p_{ni}p_{nj})^T,$$

for $1 \leq i \leq j \leq l$. Thus P^* is an $(n \times l(l+1)/2)$ matrix given by

$$P^* = \begin{bmatrix} P_1^* & P_2^* & \cdots & P_l^* \end{bmatrix}$$

where for $i = 1, \dots, l$, the matrix P_i^* is defined as

$$P_i^* = \begin{bmatrix} \mathbf{p}_i * \mathbf{p}_i & \mathbf{p}_i * \mathbf{p}_{i+1} & \cdots & \mathbf{p}_i * \mathbf{p}_l \end{bmatrix}.$$

In particular, suppose $\mathbf{v} = (v_1, \dots, v_l) \in \mathbb{F}_q^l$ then $\mathbf{v}^*$ is defined as

$$\mathbf{v}^* = (v_1 v_1, v_1 v_2, \dots, v_2 v_2, \dots, v_2 v_l, \dots, v_l v_l).$$

Proposition 4.2.9. *An ideal MSP $(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ is multiplicative if and only if the system of linear equations $\mathbf{z}M^* = \boldsymbol{\varepsilon}^*$ is solvable. Moreover, the solution $\mathbf{z}$ is a recombination vector.*

Lemma 4.2.10. *Let $G(V, E)$ be a connected graph with m vertices and n edges. For $1 \leq i \leq m$, let H_i be the reduced incidence matrix not containing the incidence vector of v_i . Then $r((H_i^T)^*) = n$ over any field, where the operation $*$ is given in Definition 4.2.8 and $r(\cdot)$ denotes the rank of a matrix. In particular, when the graph $G(V, E)$ is complete, then $(H_i^T)^*$ is nonsingular over any field.*

Proof. The result follows from the fact that each edge has two endpoints. □

Proposition 4.2.11. *Let $\mathbb{F}_q$ be a field of characteristic 2. Then the ideal MSP $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ defined in Proposition 4.2.7 is not multiplicative.*

Proof. By Proposition 4.2.9, the ideal MSP $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ is multiplicative if and only if the system of linear equations $\mathbf{z}M^* = \boldsymbol{\varepsilon}^*$ is solvable. This means that

$$r((M^*)^T) = r((M^*)^T, (\boldsymbol{\varepsilon}^*)^T).$$

Since $M = \begin{bmatrix} \mathbf{1}^T & H_m^T \end{bmatrix}$ then

$$M^* = \begin{bmatrix} \mathbf{1}^T & H_m^T & (H_m^T)^* \end{bmatrix}.$$

By Lemma 4.2.10, $r((H_m^T)^*) = n$ over $\mathbb{F}_q$, so $r((M^*)^T) = r(M^*) = n$. We can check that $r((M^*)^T, (\boldsymbol{\varepsilon}^*)^T) = n + 1$. Therefore the MSP $\mathcal{M}(\mathbb{F}_q, M, \boldsymbol{\varepsilon}, \psi)$ is not multiplicative. □

Proposition 4.2.12. *The access structure Γ_{oc} does not admit an ideal MLSSS over any field.*

Proof. First, we consider the case over a field $\mathbb{F}_q$ of characteristic not equal to 2. By Corollary 4.2.6, there is no ideal MLSSS realizing Γ_{oc} over $\mathbb{F}_q$.

Next, we consider the case over $\mathbb{F}_2$. We can deduce from Lemma 3.4.5 and Lemma 2.3.4 that if an access structure can be realized by an ideal LSSS over $\mathbb{F}_2$ then this scheme is unique over $\mathbb{F}_2$. Hence, by Proposition 4.2.7 and Proposition 4.2.11, Γ_{oc} does not admit an ideal MLSSS over $\mathbb{F}_2$.

Lastly, we look at the case over a nonbinary field $\mathbb{F}_q$ of characteristic 2. By Proposition 4.2.7, we have an ideal LSSS realizing Γ_{oc} over $\mathbb{F}_q$. Now Lemma 3.4.5 and Lemma 2.3.5 tell us that the ideal LSSS is unique. Thus, by Proposition 4.2.11, Γ_{oc} does not admit an ideal MLSSS over $\mathbb{F}_q$. $\square$

4.3 Ideal multiplicative LSSS based on graphical codes

In this section, we present the ideal multiplicative LSSS that we were able to find by using the incidence matrix and Lemma 4.2.9. We describe the corresponding access structure in terms of graphs and we show the connection to some graphical codes.

4.3.1 Access structure

Let $G(V, E)$ be a connected undirected graph with $V = \{v_1, \dots, v_m\}$ and $E = \{e_1, \dots, e_n\}$. Given $F \subseteq E$, we define $d_F(v)$ to be the number of edges in F in-

cident with the vertex v . Let Y be the collection of all $F \subseteq E$ such that $|F|$ is even and $d_F(v_i)$ is odd for $1 \leq i \leq m-1$.

Definition 4.3.1. Let K_m be the complete graph with m vertices and let n be the number of edges. For $i = 1, \dots, n$, every participant P_i is assigned to an edge. Define

$$\Gamma_{ocy} = \{A \subseteq \mathcal{P} : K_m[A] \text{ contains an odd cycle or } K_m[A] \text{ contains a set in } Y\}.$$

Then Γ_{ocy} is an access structure.

Based on the definition, Γ_{ocy} satisfies the monotone increasing property.

Example 4.3.2. Consider the complete graph K_4 (see Figure 4.3.2).

Figure 4.3.1: K_4

The minimal access structure based on odd cycles is

$$\Gamma_{oc}^- = \{\{P_1, P_3, P_5\}, \{P_1, P_2, P_4\}, \{P_2, P_3, P_6\}, \{P_4, P_5, P_6\}\}$$

while the minimal access structure based on odd cycles and Y -sets is

$$\begin{aligned} \Gamma_{ocy}^- = & \{\{P_1, P_3, P_5\}, \{P_1, P_2, P_4\}, \{P_2, P_3, P_6\}, \{P_4, P_5, P_6\}, \\ & \{P_1, P_6\}, \{P_3, P_4\}, \{P_2, P_5\}\}. \end{aligned}$$

We can check that the first one is not Q_2 while the second one is Q_2 .

Proposition 4.3.3. *The access structure Γ_{ocy} satisfies the following properties:*

- i. Let $d \geq 2$ be an integer. If $m \geq 2^d + 1$ then the access structure Γ_{ocy} is Q_d .*
- ii. If $m \geq 4$ then the access structure Γ_{ocy} is Q_2 but is not self-dual.*

Proof. Since $\Gamma_{oc} \subseteq \Gamma_{ocy}$ then the first part of the proposition follows from Proposition 4.2.2. For the second part, first we recall that self-dual access structures coincide with the minimally Q_2 access structures [37]. We claim that K_m contains a subset of edges belonging to the set Y . Indeed, if m is odd then $\{v_1v_m, v_2v_m, \dots, v_{m-1}v_m\} \in Y$ while if m is even then $\{v_1v_4, v_1v_5, \dots, v_1v_m, v_2v_3\} \in Y$. When $m \geq 5$, we have Γ_{oc} is Q_2 and $\Gamma_{oc} \subset \Gamma_{ocy}$, which means that Γ_{ocy} is not minimal. When $m = 4$, we can directly check that Γ_{ocy} is not self-dual. $\square$

4.3.2 Linear secret sharing scheme realizing Γ_{ocy}

Proposition 4.3.4. *Suppose M is an $n \times (m - 1)$ matrix over $\mathbb{F}_2$ given by*

$$M = \begin{bmatrix} \mathbf{1} + \mathbf{h}_1 \\ \mathbf{1} + \mathbf{h}_2 \\ \vdots \\ \mathbf{1} + \mathbf{h}_{m-1} \end{bmatrix}^T,$$

where $\mathbf{1}$ is the all-one vector, $\mathbf{h}_i$ is the incidence vector of vertex v_i , $1 \leq i \leq m - 1$, and $+$ is the usual binary vector addition. Then $\mathcal{M}(\mathbb{F}_2, M, \boldsymbol{\varepsilon}, \psi)$ is an ideal MSP computing the access structure Γ_{ocy} where $\boldsymbol{\varepsilon} = (1, \dots, 1)$ and ψ is a bijection.

Proof. Let $A \subseteq \mathcal{P}$ and for simplicity, we denote by $K_m[A]$ the subgraph induced by the edges corresponding to A . We shall prove that $A \in \Gamma_{ocy}$ if and only if $\boldsymbol{\varepsilon} \in \text{span}(M_A)$.

Note that $\boldsymbol{\varepsilon} \in \text{span}(M_A)$ if and only if there exists a recombination vector $\boldsymbol{w}'$ such that $\boldsymbol{\varepsilon} = \boldsymbol{w}'M_A$. Equivalently, there exists an n -dimensional vector $\boldsymbol{w}$ such that $\boldsymbol{w}M = \boldsymbol{\varepsilon}$ and with $\boldsymbol{w}_A = \boldsymbol{w}'$, $\boldsymbol{w}_{\bar{A}} = \mathbf{0}$. This means that $\boldsymbol{w}(\mathbf{1} + \boldsymbol{h}_i)^T = 1$ for $i = 1, \dots, m-1$. Now $\boldsymbol{w}(\mathbf{1} + \boldsymbol{h}_i)^T = 1$ if and only if:

$$(1) \quad \boldsymbol{w} \cdot \mathbf{1}^T = 1 \text{ and } \boldsymbol{w} \cdot \boldsymbol{h}_i^T = 0, \text{ or}$$

$$(2) \quad \boldsymbol{w} \cdot \mathbf{1}^T = 0 \quad \text{and} \quad \boldsymbol{w} \cdot \boldsymbol{h}_i^T = 1.$$

The first case means that $K_m[A]$ contains an odd cycle while the second case means that $K_m[A]$ contains a set in Y . $\square$

Consider again the extended cycle code $\widehat{\mathcal{C}}^*(K_m)$. Let $F \subseteq E$ be an element in the set Y and let $\boldsymbol{f}$ be the characteristic vector of the subgraph induced by F . We will augment $\widehat{\mathcal{C}}^*(K_m)$ by adjoining the vector $(1, \boldsymbol{f})$. We denote the resulting space by $\mathcal{D}(K_m)$. The technique of augmenting graphical codes have been considered before in [56, 65] to increase the number of codewords and to improve the decoding of graphical codes. The following proposition implies that the secret sharing scheme based on the dual of $\mathcal{D}(K_m)$ realizes the access structure Γ_{ocy} .

Proposition 4.3.5. *A parity check matrix of $\mathcal{D}(K_m)$ is given by*

$$P = \begin{bmatrix} 1 & \mathbf{1} + \boldsymbol{h}_1 \\ 1 & \mathbf{1} + \boldsymbol{h}_2 \\ \vdots & \vdots \\ 1 & \mathbf{1} + \boldsymbol{h}_{m-1} \end{bmatrix},$$

where $\boldsymbol{h}_i$ is the incidence vector of vertex v_i , $1 \leq i \leq m-1$, and $+$ is the usual binary vector addition.

Proof. First, note that it is not difficult to show that $P(1, \mathbf{f})^T = \mathbf{0}^T$. Let G be a generator matrix of $\mathcal{C}^*(K_m)$. We know that $H_m G^T = \mathbf{0}$ where H_m is the reduced incidence matrix and $\mathbf{0}$ is the zero matrix. Then a generator matrix of $\widehat{\mathcal{C}}^*(K_m)$ is given by $\widehat{G} = (\mathbf{b}^T \ G)$ where the entries in the first column $\mathbf{b}^T$ are the parity check bits of the rows of G . We can then verify that $P\widehat{G}^T = \mathbf{0}$. Therefore, P is a parity check matrix of $\mathcal{D}(K_m)$. $\square$

4.3.3 Multiplicativity

Let $\mathcal{M}(\mathbb{F}_2, M, \varepsilon, \psi)$ be the ideal MSP constructed in Proposition 4.3.4. Since the graph is complete then the number of rows of M is $n = m(m-1)/2$. Thus, M^* is an $n \times n$ matrix. Using Lemma 4.2.9, to prove that $\mathcal{M}$ is multiplicative, we need to show that $\mathbf{z}M^* = \varepsilon^*$ is solvable. In particular, we are going to look at the rank of the matrix M^* . We start with the following lemma which is not difficult to prove.

Lemma 4.3.6. *For any binary vectors $\mathbf{x}$ and $\mathbf{y}$,*

$$(\mathbf{1} + \mathbf{x}) * (\mathbf{1} + \mathbf{y}) = (\mathbf{1} + \mathbf{x} * \mathbf{y}) + \mathbf{x} + \mathbf{y}.$$

Proposition 4.3.7. *Let $\mathcal{M}(\mathbb{F}_2, M, \varepsilon, \psi)$ be the ideal MSP that computes the access structure Γ_{ocy} . Then the system of linear equations $\mathbf{z}M^* = \varepsilon^*$ is solvable if and only if $m \equiv 0, 1 \pmod{4}$.*

Proof. Applying Lemma 4.3.6, M^* will be column equivalent to the matrix $J + (H_m^T)^*$, where J is the $n \times n$ all-one matrix and H_m is the reduced incidence matrix. Let

$V = \{v_1, \dots, v_m\}$. Without loss of generality, we label the edges as follows.

$$\begin{aligned}
e_i &= v_i v_m, \quad 1 \leq i \leq m-1 \\
e_{(m-1)+i-1} &= v_1 v_i, \quad 2 \leq i \leq m-1 \\
e_{(m-1)+(m-2)+i-2} &= v_2 v_i, \quad 3 \leq i \leq m-1 \\
&\vdots \\
e_{(m-1)+(m-2)+\dots+(m-k)+i-k} &= v_k v_i, \quad k+1 \leq i \leq m-1 \\
&\vdots \\
e_{(m-1)+\dots+2+1} &= v_{m-2} v_{m-1}.
\end{aligned}$$

Then $[(H_m^T)^*]^T$ can be expressed in the form

$$\begin{bmatrix} I_{m-1} & P \\ \mathbf{0} & I_{n-m+1} \end{bmatrix}$$

where I is the identity matrix, $\mathbf{0}$ is the zero matrix and P is an $(m-1) \times (n-m+1)$ matrix, in which each row has $m-2$ nonzero entries and each column has 2 nonzero entries.

Let D be the row transformation matrix such that $D[(H_m^T)^*]^T = I_n$, then $(M^*)^T$ is row equivalent to $DJ + I_n$. Note that the system of linear equations $\mathbf{z}M^* = \boldsymbol{\varepsilon}^*$ is equivalent to the system of linear equations

$$D(M^*)^T \mathbf{z}^T = D(\boldsymbol{\varepsilon}^*)^T$$

where D is a nonsingular matrix. We consider two cases.

(1) When m is even, i.e., $m - 2$ is even, then $D(\boldsymbol{\varepsilon}^*)^T = (\boldsymbol{\varepsilon}^*)^T$, $DJ = J$, and

$$DJ + I_n = \begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}.$$

Since

$$r(DJ + I_n) = \begin{cases} n & \text{when } n \text{ is even} \\ n - 1 & \text{when } n \text{ is odd} \end{cases}$$

and $r(DJ + I_n, (\boldsymbol{\varepsilon}^*)^T) = n$, then the system of linear equations $\boldsymbol{z}M^* = \boldsymbol{\varepsilon}^*$ is solvable if and only if n is even. Now, n is even if and only if $m \equiv 0 \pmod{4}$.

(2) When m is odd, i.e., $m - 2$ is odd,

$$DJ = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix},$$

where there are $m - 1$ all-zero rows and there are $n - m + 1$ all-one rows.

Similarly,

$$D(\boldsymbol{\varepsilon}^*)^T = (0 \ 0 \ \dots \ 0 \ 1 \ 1 \ \dots \ 1)^T$$

where the vector at the right hand of the equation contains $m - 1$ zeros. Thus the system of linear equations $\mathbf{z}M^* = \boldsymbol{\varepsilon}^*$ is equivalent to the system of linear equations

$$\begin{bmatrix} I_{m-1} & \mathbf{0} \\ J_{(n-m+1) \times (m-1)} & E \end{bmatrix} \mathbf{z}^T = D(\boldsymbol{\varepsilon}^*)^T,$$

where E is an $(n - m + 1) \times (n - m + 1)$ square matrix given by

$$E = \begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 0 \end{bmatrix}.$$

Using a similar argument as in the previous case, we can see that when m is odd, the system of linear equations $\mathbf{z}M^* = \boldsymbol{\varepsilon}^*$ is solvable if and only if n is even.

This means that $m \equiv 1 \pmod{4}$.

□

Corollary 4.3.8. *Let $\mathcal{M}(\mathbb{F}_2, M, \boldsymbol{\varepsilon}, \psi)$ be the ideal MSP that computes the access structure Γ_{ocy} . Then $\mathcal{M}$ is multiplicative if and only if $m \equiv 0, 1 \pmod{4}$.*

4.4 Conclusion

In this chapter, we studied the access structure and multiplicativity of linear secret sharing schemes based on certain binary codes from complete graphs. We have shown that the access structure based on odd cycles does not admit an ideal multiplicative

LSSS. We also obtained some ideal multiplicative LSSS from the dual of an augmented extended cycle code.

One interesting open problem to consider is the multiplicativity of LSSS based on codes from non-complete but connected graphs. Another interesting problem is the possibility of obtaining strongly multiplicative LSSS based on codes from graphs.

Chapter 5

A generalization of SSS based on linear codes

In this chapter, we consider a generalization of the SSS based on linear codes studied by Karnin, Greene and Hellman, by Bertilsson and Ingemarsson, and by Massey. Unlike in the classical case, there are only a few works on this general SSS. In [48], Dougherty, Mesnager and Solé analyzed the access structure of the SSS based on self-dual codes. They observed that the design properties and joint weight enumerators of self-dual codes can be used to get the weight distribution of the codewords in the dual code with first coordinate equal to 1. This weight distribution also gives the number of authorized sets of different sizes.

Our main objective in this chapter is to extend their techniques and results to the case of general SSS based on linear codes. The generalized scheme construction is presented in Section 5.1. In Section 5.2, we characterize the authorized groups using the dual code, extending a result by Massey. In the remaining sections, we analyze

the access structure of general SSS based on binary self-dual codes by extending the techniques used in [48]. The material in this chapter can be found in [40].

5.1 General SSS based on linear codes

Let $\mathcal{P} = \{P_1, \dots, P_n\}$ be the set of participants. The set of all possible secrets is $\mathcal{S} = \mathbb{F}_q^l$ where $l \geq 2$ and the set of all possible shares for P_i is $\mathcal{S}_i = \mathbb{F}_q$. Suppose we want to share the secret $\mathbf{s} = (s_1, s_2, \dots, s_l) \in \mathbb{F}_q^l$. Let $\mathcal{C}$ be an $[l+n, k, d]$ linear code over $\mathbb{F}_q$ with $d > l$. Consider a generator matrix

$$G = \begin{bmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_l & \mathbf{g}_{l+1} & \cdots & \mathbf{g}_{l+n} \end{bmatrix}$$

where $\mathbf{g}_i$ is the i th column. To generate the shares, the dealer picks randomly a vector $\mathbf{u} \in \mathbb{F}_q^k$ such that

$$\mathbf{u}\mathbf{g}_i = s_i \text{ for } 1 \leq i \leq l.$$

The dealer then computes the corresponding codeword

$$\mathbf{c} = (c_1, c_2, \dots, c_l, c_{l+1}, \dots, c_{l+n}) = \mathbf{u}G$$

where the first l coordinates are the components of the secret. Now the share of P_i is c_{l+i} for $i = 1, \dots, l$. The information rate of the scheme is l . Thus, the scheme is non-perfect.

We note that when $l = 1$ then we have Scheme II in Chapter 3, Section 3.5. This construction is not new as it was already considered in [67, 85] for threshold schemes and in [20] for the general case. It is also related to the generalized vector space construction in [109].

In [20], Blakley and Kabatianskii proved the following theorem that gives a characterization of the amount of information that can be obtained by a subset of participants.

Theorem 5.1.1. *Let $B = \{P_{i_1}, P_{i_2}, \dots, P_{i_m}\} \subseteq \mathcal{P}$ and define*

$$J = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_l\} \cap \text{span}\{\mathbf{g}_{i_1}, \mathbf{g}_{i_2}, \dots, \mathbf{g}_{i_m}\}.$$

i. $B \in \Gamma$, i.e., B can uniquely determine the secret, if and only if

$$J = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_l\}.$$

ii. $B \in \Delta$, i.e., B cannot obtain any information about the secret, if and only if

$$J = \mathbf{0}.$$

iii. $B \in (\Gamma \cup \Delta)^c$, i.e., B can obtain partial information about the secret, if and only if

$$\mathbf{0} \subset J \subset \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_l\}.$$

5.2 Access Structure

We study in this section the access structure of the general SSS based on linear codes.

We start by giving a characterization of the authorized groups using the dual code that generalizes Theorem 3.5.2.

Proposition 5.2.1. *Let $\mathcal{C}$ be an $[l+n, k, d]$ linear code over $\mathbb{F}_q$ and let B be a subset of $\mathcal{P}$ such that $B = \{P_{i_1}, \dots, P_{i_m}\}$. In the general SSS based on $\mathcal{C}$ the participants in B can determine the secret $\mathbf{s}$ if and only if there exist codewords $\mathbf{v}_j \in \mathcal{C}^\perp$, $1 \leq j \leq l$, satisfying the following conditions:*

i. The subvector of $\mathbf{v}_j$ consisting of its first l coordinates is equal to the j th unit vector $\mathbf{e}_j$ in $\mathbb{F}_q^l$.

ii. $\text{supp}(\mathbf{v}_j) \subseteq \{j, i_1, \dots, i_m\}$.

Proof. Suppose there exist codewords $\mathbf{v}_j \in \mathcal{C}^\perp$, $1 \leq j \leq l$, satisfying conditions (i) and (ii). For $j = 1, \dots, l$, we have

$$\mathbf{s} \cdot \mathbf{v}_j = c_j + \sum_{r=1}^m \alpha_{jr} c_{i_r} = 0$$

for some constants α_{jr} , $1 \leq r \leq m$, which are not all zero. Hence, the secret $\mathbf{s}$ can be determined as a linear combination of the shares of participants in B .

Suppose the participants in B can determine the secret. Then for each $j = 1, \dots, l$, we have an equation of the form

$$c_j = \sum_{r=1}^m \beta_{jr} c_{i_r}$$

for some constants β_{jr} , $1 \leq r \leq m$, which are not all zero. The equation can be rewritten as

$$(c_1, c_2, \dots, c_l, c_{l+1}, \dots, c_{l+n}) \cdot (\mathbf{e}_j, 0, \dots, -\beta_{j1}, \dots, -\beta_{jm}, 0, \dots, 0) = 0.$$

Now the codewords $(\mathbf{e}_j, 0, \dots, -\beta_{j1}, \dots, -\beta_{jm}, 0, \dots, 0)$ are in $\mathcal{C}^\perp$ and satisfy conditions (i) and (ii). $\square$

Example 5.2.2. Let $\mathcal{C}_1$ be the $[8, 3, 4]$ -linear code over $\mathbb{F}_3$ with generator matrix

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

We consider the scheme based on the dual of $\mathcal{C}_1$ with $l = 2$ (so we have 6 participants). Applying the proposition, we can verify that the access structure consists of 4 groups of size 5 and 1 group of size 6.

Example 5.2.3. Consider the scheme based on the $[8, 4, 4]$ extended binary Hamming code with $l = 3$. In this case, we have a total of 5 participants. There are 4 groups of size 4 and 1 group of size 5 in the access structure.

In [33], it was shown that any group of size at most $d^\perp - l - 1$ has no information about the secret and any group of size at least $n + l - d + 1$ can recover the secret. Here we show that no group of size at most $d_l^\perp - l - 1$ is in the access structure, where $d_l^\perp$ is the l th generalized Hamming weight of $\mathcal{C}$.

Corollary 5.2.4. Any group of $d_l^\perp - l - 1$ or less participants is not in the access structure where $d_l^\perp$ is the l th generalized Hamming weight of $\mathcal{C}^\perp$.

Proof. The l th generalized Hamming weight of a linear code is the minimum support of its subcodes of dimension l . A minimal access group $B = \{P_{i_1}, \dots, P_{i_m}\}$ corresponds to an $[l + n, l]$ subcode $\mathcal{D}$ of $\mathcal{C}^\perp$ such that $\text{supp}(\mathcal{D}) = \{1, \dots, l, i_1, \dots, i_m\}$. Hence, $m \geq d_l^\perp - l$. $\square$

In general, it is not easy to determine $d_l^\perp$ for $l \geq 2$. In the case of binary linear codes, we can show that the size of an access group is at least $\frac{3}{2}(d^\perp - l)$, where $d^\perp$ is the minimum weight of $\mathcal{C}^\perp$. This bound is weaker than the one given by $d_l^\perp$, but easier to calculate.

Corollary 5.2.5. Consider the general SSS based on a binary linear code $\mathcal{C}$. If $l \geq 2$ then any group of $\frac{3}{2}(d^\perp - l) - 1$ or less participants is not in the access structure.

Proof. As in the proof of Corollary 5.2.4, a minimal access group of size m corresponds to an $[l + n, l]$ subcode $\mathcal{D}$ of $\mathcal{C}^\perp$ whose support has size $l + m$. Moreover, deleting the first l coordinates of $\mathcal{D}$ as well as those coordinates which are not in its support yields a binary $[m, l]$ code of minimum weight at least $d^\perp - l$. Recall that $A_2(n, d)$ is the maximum size of a (not necessarily linear) binary code of length n and minimum weight at least d . The above yields $A_2(m, d^\perp - l) \geq 2^l > 2$. On the other hand, it is well-known that $A_2(n, d) \leq 2$ whenever $n \leq \frac{3}{2}d - 1$. This yields $m \geq \frac{3}{2}(d^\perp - l)$. $\square$

Proposition 5.2.6. *When all participants come together and attempt to determine the secret, at most $\lfloor (d - 1)/2 \rfloor$ invalid shares can be identified and corrected.*

Proof. The secret and shares form a codeword of a linear code $\mathcal{C}$ with minimum distance d . Hence, at most $\lfloor (d - 1)/2 \rfloor$ errors can be corrected. $\square$

5.3 Relation between access structure and joint weight enumerator

We describe the connection between the g -fold joint weight enumerator and the access structure. The following definition is taken from [47].

Definition 5.3.1. *Let $A_1, A_2, \dots, A_g$ be codes of length n over $\mathbb{F}_q$. The g -fold joint weight enumerator of $A_1, A_2, \dots, A_g$ is defined as follows:*

$$\mathcal{J}_{A_1, A_2, \dots, A_g}(x_a; a \in \mathbb{F}_2^g) = \sum_{\mathbf{c}_1 \in A_1, \dots, \mathbf{c}_g \in A_g} \prod_{a \in \mathbb{F}_2^g} x_a^{n_a(\mathbf{c}_1, \dots, \mathbf{c}_g)},$$

where $\mathbf{c}_j = (c_{j1}, \dots, c_{jn})$, $n_a(\mathbf{c}_1, \dots, \mathbf{c}_g) = |\{i | a = (\overline{c_{1i}}, \dots, \overline{c_{gi}})\}|$, and $\overline{c_{ji}} = 1$ if $c_{ji} \neq 0$ and $\overline{c_{ji}} = 0$ if $c_{ji} = 0$. Here $(x_a; a \in \mathbb{F}_2^g)$ is a 2^g -tuple of variables with $\mathbb{F}_2^g$, that is,

$$(x_{00\dots 0}, x_{00\dots 1}, \dots, x_{11\dots 1}).$$

When $g = 2$ then we have the *joint weight enumerator* [77]. The joint weight enumerator of a code with itself is called the *biweight enumerator*. The joint weight enumerator is a generalization of the weight enumerator.

From this point to the rest of this chapter, we only deal with general secret sharing schemes based on binary linear codes. First we consider the case $l = 2$, i.e. the secret $\mathbf{s} = (s_1, s_2)$. For simplicity, we use the corresponding decimal representation of the subscripts of the variables in the g -fold joint weight enumerator. Let $T_1 = \{1\}$ and $T_2 = \{2\}$ with indicator vectors 1_{T_1} and 1_{T_2} respectively. Consider the 4-fold joint weight enumerator $\mathcal{J}_{1_{T_1}, 1_{T_2}, \mathcal{C}^\perp, \mathcal{C}^\perp}(x_a)$ where $a \in \mathbb{F}_2^4$. We are interested in the coefficient of $x_{10}x_5$. The coefficient is a polynomial in $x_0x_1x_2x_3$ and it gives information on the number and supports of pairs of codewords $\mathbf{u}, \mathbf{v} \in \mathcal{C}^\perp$ whose first two coordinates are $(u_1, 0)$ and $(0, v_2)$ respectively, where u_1 and v_2 are both non-zero. In general, for secrets of length l we use the $2l$ -fold joint weight enumerator $\mathcal{J}_{1_{T_1}, \dots, 1_{T_l}, \mathcal{C}^\perp, \dots, \mathcal{C}^\perp}(x_a; a \in \mathbb{F}_2^{2l})$ where $a \in \mathbb{F}_2^{2l}$. The following theorem generalizes a result in [48] where Jacobi polynomials were used.

Theorem 5.3.2. *Let X_1 be the subset of $\mathbb{F}_2^{2l}$ consisting of all vectors whose first l coordinates are zero and let*

$$X_2 := \{(\mathbf{e}_j, \mathbf{e}_j) \mid j \in \{1, \dots, l\}\},$$

where $\mathbf{e}_j \in \mathbb{F}_2^l$ is the j th unit vector. Then the coefficient of $\prod_{a \in X_2} x_a$ in

$$\mathcal{J}_{1_{T_1}, \dots, 1_{T_l}, \mathcal{C}^\perp, \dots, \mathcal{C}^\perp}(x_a; a \in \mathbb{F}_2^{2l})$$

is a polynomial $p(x_a; a \in X_1)$. Identify X_1 with $\{0, \dots, 2^l - 1\}$ via the binary number representation and write

$$p = \sum_{\mu \in \mathbb{N}_0^{2^l}} c_\mu \prod_{a \in X_1} x_a^{\mu_a}.$$

Then the number $M_{\mathcal{C}}(m)$ of groups of size m in the access structure of the scheme based on $\mathcal{C}$ satisfies

$$M_{\mathcal{C}}(m) \leq \sum_{\mu} c_\mu,$$

where the sum is over all μ with $\sum_{i=1}^{2^l-1} \mu_i = m$. Moreover, if $m < \frac{3}{2}d^\perp - 1$ then equality holds.

Proof. The sum of the coefficients c_μ , where $\sum_{i=1}^{2^l-1} \mu_i = m$, equals the number of tuples $(\mathbf{v}_1, \dots, \mathbf{v}_l)$ of elements of $\mathcal{C}^\perp$ such that the projection of $\mathbf{v}_j$ onto the first l coordinates is the j th unit vector in $\mathbb{F}_2^l$, and

$$|\cup_{j=1}^l \text{supp}(\mathbf{v}_j) \cap \{l+1, \dots, l+n\}| = m.$$

Hence due to Proposition 5.2.1, every such tuple determines a group in the access structure of the scheme based on $\mathcal{C}$, and every minimal access group occurs as a union of supports of such a tuple. However, in general there may be different tuples of codewords that correspond to the same access group. In this situation, there exists a tuple $(\mathbf{v}_1, \dots, \mathbf{v}_l)$ as above and an element $\mathbf{c} \in \mathcal{C}^\perp$ such that

$$\text{supp}(\mathbf{c}) \subseteq \cup_{j=1}^l \text{supp}(\mathbf{v}_j) \cap \{l+1, \dots, l+n\}.$$

Then for any $j \in \{1, \dots, l\}$,

$$|\text{supp}(\mathbf{c}) \cap \text{supp}(\mathbf{v}_j) \cap \{l+1, \dots, l+n\}| \geq \text{wt}(\mathbf{c}) + \text{wt}(\mathbf{v}_j) - 1 - m$$

and hence

$$\begin{aligned}
d^\perp &\leq \text{wt}(\mathbf{c} + \mathbf{v}_j) \\
&\leq 1 + m - (\text{wt}(\mathbf{c}) + \text{wt}(\mathbf{v}_j) - 1 - m) \\
&\leq 2m + 2 - 2d^\perp,
\end{aligned}$$

which yields $m \geq \frac{3}{2}d^\perp - 1$. Hence if $m < \frac{3}{2}d^\perp - 1$ then the sum of the coefficients c_μ with $\sum_{i=1}^{2^l-1} \mu_i = m$ equals the number of access groups of size m . $\square$

If $\mathcal{C}$ is self-orthogonal then there exists a weaker condition than the one in Theorem 5.3.2 under which the number of access groups of size m can be read off from the $2l$ -fold joint weight enumerator. To state this condition, we need the notion of the *code extension enumerator* below.

Definition 5.3.3. *Let $\mathcal{D}$ be a linear self-orthogonal $[n, k, d]$ code. For a vector $\mathbf{c}$ not in $\mathcal{D}$, we denote by $\langle \mathbf{c}, \mathcal{D} \rangle$ the code generated by $\mathbf{c}$ and $\mathcal{D}$. The code extension enumerator is the complex polynomial*

$$P_{\mathcal{D}}(t) = \sum_{\mathbf{c}} t^{d(\langle \mathbf{c}, \mathcal{D} \rangle)},$$

where the sum is over a system of representatives of $\mathcal{D}^\perp/\mathcal{D}$.

Clearly $\deg(P_{\mathcal{D}}) \leq d$, and a summand $t^{d'}$ in $P_{\mathcal{D}}(t)$ gives rise to a linear self-orthogonal $[n, k+1, d']$ -code.

Now consider the general SSS based on a binary self-orthogonal linear code $\mathcal{C}$ and let $(\mathbf{v}_1, \dots, \mathbf{v}_l)$ be a tuple of elements of $\mathcal{C}^\perp$ giving rise to an access group of size m , as in Proposition 5.2.1. Let $\mathcal{D}$ be the linear code generated by all the $\mathbf{v}_j$, where the columns where all the $\mathbf{v}_j$ are zero are deleted. Then $\mathcal{D}$ is a self-orthogonal $[l+m, l]$ code of minimum distance at least $d^\perp$.

Assume that there exists another tuple of elements of $\mathcal{C}^\perp$ leading to the same access group, i.e., in Theorem 5.3.2, we have strict inequality for $M_{\mathcal{C}}(m)$. Then there exists a nonzero element $\mathbf{c} \in \mathcal{C}^\perp$ with

$$\text{supp}(\mathbf{c}) \subseteq \cup_{j=1}^l \text{supp}(\mathbf{v}_j) \cap \{l+1, \dots, l+n\}.$$

Let $(\mathbf{c})' \in \mathbb{F}_2^{l+m}$ be obtained from $\mathbf{c}$ by deleting the coordinates where all the $\mathbf{v}_j$ are zero. Then $\langle (\mathbf{c})', \mathcal{D} \rangle$ has minimum weight at least $d^\perp$, hence gives rise to a summand $t^{d(\langle (\mathbf{c})', \mathcal{D} \rangle)}$ in $P_{\mathcal{D}}(t)$, where $d(\langle (\mathbf{c})', \mathcal{D} \rangle) \geq d^\perp$. This yields

Corollary 5.3.4. *Consider the general SSS based on a binary self-orthogonal linear code $\mathcal{C}$ and let $\mathcal{T}$ be the set of all tuples in $\mathcal{C}^\perp$ that give rise to an access group of size m (cf. Proposition 5.2.1). For a tuple $(\mathbf{v}_1, \dots, \mathbf{v}_l) \in \mathcal{T}$, let $\mathcal{D}(\mathbf{v}_1, \dots, \mathbf{v}_l)$ be the code generated by all the $\mathbf{v}_j$, in which the columns where all the $\mathbf{v}_j$ are zero are deleted. If for all such tuples, all monomials in $P_{\mathcal{D}(\mathbf{v}_1, \dots, \mathbf{v}_l)}(t)$ (except for the monomial corresponding to $0 \in \mathcal{D}^\perp / \mathcal{D}$) have degree less than $d^\perp$ then equality holds in Theorem 5.3.2, i.e. the number of groups of size m in the access structure of the scheme based on $\mathcal{C}$ can be read off from $\mathcal{J}_{1_{T_1}, \dots, 1_{T_l}, \mathcal{C}^\perp, \dots, \mathcal{C}^\perp}$.*

5.4 Binary self-dual codes

In this section, we focus on schemes based on binary self-dual codes and the case $l = 2$. The *automorphism group* of a binary code $\mathcal{C}$ is the set of all permutations of coordinates which map the code to itself. The automorphism group is said to be *t-transitive* if for each pair of subsets (of coordinates) E_1, E_2 with t elements, there is a permutation in the group which maps E_1 to E_2 . When $t = 1$, we use the term

transitive automorphism group.

Based on the previous section, we use $\mathcal{J}_{1_{T_1}, 1_{T_2}, \mathcal{C}, \mathcal{C}}(x_0, \dots, x_{15})$ and determine the coefficient of $x_{10}x_5$. Let us denote this coefficient by Z . Under some conditions, we can determine Z using the biweight enumerator of $\mathcal{C}$.

Proposition 5.4.1. *Let $\mathcal{C}$ be an $[n, k, d]$ binary self-dual code. If $\mathcal{C}$ has a 2-transitive automorphism group then*

$$\begin{aligned} Z &= \frac{1}{n(n-1)} \frac{\partial^2}{\partial x_2 \partial x_3} \mathcal{J}_{\mathcal{C}, \mathcal{C}}(x_0, x_1, x_2, x_3) \\ &= \frac{1}{n(n-1)} \frac{\partial^2}{\partial x_3 \partial x_2} \mathcal{J}_{\mathcal{C}, \mathcal{C}}(x_0, x_1, x_2, x_3). \end{aligned}$$

Proof. The first part of the proof is taken from [58]. We can write the biweight enumerator as

$$\mathcal{J}_{\mathcal{C}, \mathcal{C}}(x_0, x_1, x_2, x_3) = \sum A_{i,j,k,l} x_0^i x_1^j x_2^k x_3^l$$

where $A_{i,j,k,l}$ is the number of pairs of codewords with $n_{00} = i, n_{01} = j, n_{10} = k, n_{11} = l$. For a given coefficient $A_{i,j,k,l}$ and coordinate position h , let $N_h(i, j, k, l)$ be the set of all pairs of codewords in $\mathcal{C}$ which contribute to $A_{i,j,k,l}$ and with 01 pattern at h . It follows that

$$\sum_{h=1}^n |N_h(i, j, k, l)| = j A_{i,j,k,l}$$

since any pair in N_h has j positions with the 01 pattern. Since the automorphism group is transitive then $|N_h(i, j, k, l)|$ is independent of h . Thus,

$$|N_h(i, j, k, l)| = \frac{j}{n} A_{i,j,k,l}$$

and in particular,

$$|N_2(i, j, k, l)| = \frac{j}{n} A_{i,j,k,l}.$$

Let $N'_h(i, j, k, l)$ be the set of all pairs of codewords in $N_2(i, j, k, l)$ with 10 pattern at position h . Using the arguments above and since the automorphism group is 2-transitive, then $|N'_h(i, j, k, l)|$ is independent of h and

$$\begin{aligned} |N'_h(i, j, k, l)| &= \frac{k}{n-1} |N_2(i, j, k, l)| \\ &= \frac{kj}{n(n-1)} A_{i,j,k,l}. \end{aligned}$$

The proposition now follows. $\square$

Since the following examples deal with self-dual codes, we shall remark the following.

Proposition 5.4.2. *In the general SSS with $l = 2$ based on a binary self-dual code $\mathcal{C}$, the size of every minimal group in the access structure is even.*

Proof. A minimal access group of size m in the access structure corresponds to a pair $(\mathbf{v}_1, \mathbf{v}_2)$ of words in $\mathcal{C}^\perp = \mathcal{C}$ such that $\mathbf{v}_1 = (1, 0, \dots)$ and $\mathbf{v}_2 = (0, 1, \dots)$ and

$$m = |(\text{supp}(\mathbf{v}_1) \cup \text{supp}(\mathbf{v}_2)) - \{1, 2\}|.$$

The latter equals

$$\text{wt}(\mathbf{v}_1) - 1 + \text{wt}(\mathbf{v}_2) - 1 - |\text{supp}(\mathbf{v}_1) \cap \text{supp}(\mathbf{v}_2)|.$$

Since $\mathcal{C}$ is self-dual, the weight of every word in $\mathcal{C}$ is even. Moreover, the parity of $|\text{supp}(\mathbf{v}_1) \cap \text{supp}(\mathbf{v}_2)|$ equals the inner product of $\mathbf{v}_1$ with $\mathbf{v}_2$, hence is zero as well.

Hence m is even. $\square$

Example 5.4.3. *The automorphism group of the $[8, 4, 4]$ extended Hamming code is*

2-transitive and its biweight enumerator is

$$\begin{aligned}\mathcal{J}_{c,c}(x_0, x_1, x_2, x_3) = & x_3^8 + 14x_2^4x_3^4 + x_2^8 + 14x_3^4x_1^4 + 14x_2^4x_1^4 + x_1^8 + 168x_0^2x_1^2x_2^2x_3^2 \\ & + 14x_3^4x_0^4 + 14x_2^4x_0^4 + 14x_1^4x_0^4 + x_0^8.\end{aligned}$$

We obtain $Z = 4x_1^3x_2^3 + 12x_0^2x_1x_2x_3^2$. When $l = 2$, the total number of participants is 6. Since $\frac{3}{2}d^\perp - 1 = 5$, we can read off the number of access groups of size 4 as 12. The only other access group is the one formed by all participants.

Example 5.4.4. The biweight enumerator of the $[24, 12, 8]$ Golay code g_{24} was computed in [77] and it is known that the automorphism group of this code is 5-transitive. Applying the proposition above, we obtain

$$\begin{aligned}Z = & 6160x_0^{12}x_1^3x_2^3x_3^4 + 22176x_0^{10}x_1^5x_2^5x_3^2 + 7392x_0^{10}x_1^5x_2x_3^6 + 7392x_0^{10}x_1x_2^5x_3^6 \\ & + 2640x_0^8x_1^7x_2^7 + 73920x_0^8x_1^7x_2^3x_3^4 + 73920x_0^8x_1^3x_2^7x_3^4 + 36960x_0^8x_1^3x_2^3x_3^8 \\ & + 36960x_0^6x_1^9x_2^5x_3^2 + 12320x_0^6x_1^9x_2x_3^6 + 36960x_0^6x_1^5x_2^9x_3^2 + 266112x_0^6x_1^5x_2^5x_3^6 \\ & + 7392x_0^6x_1^5x_2x_3^{10} + 12320x_0^6x_1x_2^9x_3^6 + 7392x_0^6x_1x_2^5x_3^{10} + 18480x_0^4x_1^{11}x_2^3x_3^4 \\ & + 147840x_0^4x_1^7x_2^7x_3^4 + 73920x_0^4x_1^7x_2^3x_3^8 + 18480x_0^4x_1^3x_2^{11}x_3^4 + 73920x_0^4x_1^3x_2^7x_3^8 \\ & + 6160x_0^4x_1^3x_2^3x_3^{12} + 36960x_0^2x_1^9x_2^5x_3^6 + 36960x_0^2x_1^5x_2^9x_3^6 + 22176x_0^2x_1^5x_2^5x_3^{10} \\ & + 176x_1^{15}x_2^7 + 672x_1^{11}x_2^{11} + 176x_1^7x_2^{15} + 2640x_1^7x_2^7x_3^8.\end{aligned}$$

For the general SSS based on g_{24} with secret length $l = 2$, the number of groups in the access structure of size $m = 10$ can be read off from Z as 6160 due to Theorem 5.3.2, since $10 < \frac{3}{2}d^\perp - 1 = 11$. For every tuple $(\mathbf{v}_1, \mathbf{v}_2)$ giving rise to an access group of size $m = 12$, we can compute $P_{D(\mathbf{v}_1, \mathbf{v}_2)}(t)$ explicitly, using the information on the pairs of codewords that is given by Z . It turns out that in all the cases, all monomials

have degree less than 8, hence due to Corollary 5.3.4, the number of access groups of size 12 equals 36960.

5.5 Invariant theory

Suppose $\mathcal{C}$ is an $[n, k, d]$ binary self-dual code. We shall apply invariant theory in describing the access structure, similar to what was done in [48]. We consider the case $l = 2$. Thus, we shall look at the 4-fold joint weight enumerator $\mathcal{J}_{1_{T_1}, 1_{T_2}, \mathcal{C}, \mathcal{C}}(x_a)$ where $a \in \mathbb{F}_2^4$.

If all the codewords of $\mathcal{C}$ have weights divisible by 4 then we have a Type II code. Otherwise, we have a Type I code. In [77], it was shown that the biweight enumerator of a Type I code is invariant under the group G_1 generated by all permutation matrices, all 16 matrices $\text{diag}(\pm 1, \pm 1, \pm 1, \pm 1)$, and

$$T_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

The biweight enumerator of a Type II code is invariant under the group G_2 generated by G_1 and $T_2 = \text{diag}(1, i, 1, i)$ [58].

Let G stand for G_1 or G_2 depending on the type of code we are dealing with. Following the arguments in [58] and [77, Section III], and using the MacWilliams theorem in [47], we can verify that $\mathcal{J}_{1_{T_1}, 1_{T_2}, \mathcal{C}, \mathcal{C}}(x_a)$ is left invariant by every element

of G acting simultaneously on the following sets of variables:

$$V_1 = \{x_0, x_1, x_2, x_3\}, V_2 = \{x_4, x_5, x_6, x_7\},$$

$$V_3 = \{x_8, x_9, x_{10}, x_{11}\}, V_4 = \{x_{12}, x_{13}, x_{14}, x_{15}\}.$$

Hence, $\mathcal{J}_{1_{T_1}, 1_{T_2}, \mathcal{C}, \mathcal{C}}(x_a)$ is a simultaneous invariant for the diagonal action of G . As a consequence, we can extend the results in [48] regarding the Molien series. Note that the exponents of the variables in V_4 are always zero, hence we can just consider the remaining three sets. The vector space of invariants that we are going to use is $\mathbb{C}[x_a]_{i,j,k}^G$ where $x_a \in \mathbb{F}_2^4 \setminus V_4$ and i, j, k are the total degrees of the variables in V_1, V_2, V_3 respectively. The corresponding generalized Molien series [105] is given by

$$\begin{aligned} \Phi_G(r, s, t) &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \dim(\mathbb{C}[x_a]_{i,j,k}^G) \\ &= \frac{1}{|G|} \sum_{g \in G} \frac{1}{\det(I - rg) \det(I - sg) \det(I - tg)}. \end{aligned}$$

Based on the previous section, we are interested in $\dim(\mathbb{C}[x_a]_{r,1,1}^G)$. Its generating function in the variable r is given by

$$F_G(r) = \left. \frac{\partial}{\partial s \partial t} \Phi_G(r, s, t) \right|_{(s,t)=(0,0)}.$$

Using MAGMA [25], we obtain the following for Type I:

$$\begin{aligned} F_G(r) &= (r^{20} + r^{16} - 2r^{14} + 2r^{12} + r^{10} + r^8 - r^6 + 1) \\ &\quad / (r^{32} - 2r^{30} + 2r^{28} - 4r^{26} + 5r^{24} - 4r^{22} + 6r^{20} - 6r^{18} + 4r^{16} - 6r^{14} \\ &\quad + 6r^{12} - 4r^{10} + 5r^8 - 4r^6 + 2r^4 - 2r^2 + 1). \end{aligned}$$

For Type II we have

$$\begin{aligned} F_G(r) &= (4r^{62} + 4r^{54} + 5r^{46} + 6r^{38} + 7r^{30} + 3r^{22} + 2r^{14} + r^6) \\ &\quad / (r^{96} - r^{88} - 2r^{72} + 2r^{64} - r^{56} + 2r^{48} - r^{40} + 2r^{32} - 2r^{24} - r^8 + 1). \end{aligned}$$

5.6 Conclusion

We considered in this chapter an extension of secret sharing schemes based on linear codes where the secret is a vector of field elements. First, we have characterized the authorized sets in terms of codewords in the dual of the underlying code. Then we have shown an application of the g -fold joint weight enumerator to count the number of authorized sets of certain sizes.

We note that we were only able to obtain a partial count of the authorized sets. Thus, the complete description of access structures is still an open problem. It would be worthwhile also to replace symmetry properties (group transitivity) by regularity properties (combinatorial properties) in Theorem 5.4.1.

Chapter 6

The maximum number of minimal codewords in binary codes

This chapter focuses on the minimal codewords in binary linear codes. Minimal codewords (defined in Chapter 2) have been used in decoding algorithms and in secret sharing schemes. In [83], Massey showed that in a secret sharing schemes based on linear code, there is a one-to-one correspondence between the minimum authorized groups and certain minimal codewords of the dual code with a nonzero first coordinate. For a proof of this important result, we refer the reader to Theorem 3.5.2 in Chapter 3, Section 3.5.

Given a linear code, it is important to know its minimal codewords. However, in general, this is a very hard task and it is completed only for a few classes of codes. In this chapter, $\mathcal{C}$ denotes a binary linear code of length n and dimension k . We define $W(\mathcal{C})$ to be the number of its minimal codewords. We are interested with the following question:

What is the maximum number of minimal codewords a code $\mathcal{C}$ can have?

Answering this question may give us an idea on which access structures can be realized by a scheme based on linear codes. For example, suppose we are given the number of minimal groups in an access structure. If this number exceeds the maximal number for some n and k , then we know that this access structure cannot be realized by an SSS based on an $[n, n - k]$ binary linear code.

This question have been considered before in the setting of graphs and matroids. If $\mathcal{C}$ is the cycle code of a connected graph G with m vertices and n edges then $\mathcal{C}$ is a binary $[n, n - m + 1]$ -linear code. The minimal codewords of $\mathcal{C}$ are exactly the characteristic vectors of the cycles in G . Hence, the question above can be stated in terms of cycles in connected graphs. This problem was studied by Entringer and Slater in [49] where they showed that a connected graph with m vertices and n edges cannot have 2^{n-m+1} cycles.

In [46], Dósa, Szalkai and Laflamme investigated the maximum number of circuits of matroids given the size and rank. They proved an upper bound and showed that only the uniform matroids satisfy this bound. The question above is a special case of this problem since the minimal codewords of $\mathcal{C}$ are the circuits of the matroid represented by the parity-check matrix of $\mathcal{C}$.

In Section 6.1, we discuss a number of general upper bounds and lower bounds on the quantity in question. We give explicit bounds and exact values in Section 6.2. In Section 6.3, we present an asymptotic analysis. We determine the asymptotic value of the maximum number of minimal codewords for cycle codes. The material in this chapter can be found in [3, 4].

6.1 Bounds

Let $\mathbf{C}[n, k]$ denote the set of all codes of length n and dimension k . We define

$$W(n, k) = \max\{W(\mathcal{C}) : \mathcal{C} \in \mathbf{C}[n, k]\},$$

that is, $W(n, k)$ is the maximum of $W(\mathcal{C})$ over all $[n, k]$ -codes.

6.1.1 Upper bounds

We note that if $\mathcal{C}$ is of dimension k then $W(\mathcal{C})$ is at most $2^k - 1$. Hence,

$$W(n, k) \leq 2^k - 1.$$

We will call this bound the *trivial upper bound*. It implies the bound for the number of cycles in a connected graph by Entringer and Slater.

A binary linear code $\mathcal{C}$ is *intersecting* if the supports of any two nonzero codewords are not disjoint. The trivial upper bound is met with equality for intersecting codes.

Lemma 6.1.1. *A binary linear code $\mathcal{C}$ of dimension k is intersecting if and only if $W(\mathcal{C}) = 2^k - 1$.*

Proof. Suppose $\mathcal{C}$ is not intersecting. Then there exist two codewords $\mathbf{u}, \mathbf{v}$ whose supports are disjoint. It follows that the sum $\mathbf{u} + \mathbf{v}$ is nonzero and is not a minimal codeword. Thus, $W(\mathcal{C}) < 2^k - 1$. Conversely, suppose $\mathcal{C}$ has a non-minimal codeword $\mathbf{u}$. Then there exists a codeword $\mathbf{v}$ such that $\text{supp}(\mathbf{v}) \subset \text{supp}(\mathbf{u})$. This implies that the codewords $\mathbf{v}$ and $\mathbf{u} + \mathbf{v}$ have disjoint supports. $\square$

The following theorem was proved in [46] and we present here a short independent proof.

Theorem 6.1.2. *If M is a matroid on n elements and of rank r , then M has at most*

$$\binom{n}{r+1}$$

circuits (that is, minimal dependent sets).

Proof. The proof is by induction on n . Let B be a basis of M . If $B = M$, then $r = n$ and the number of circuits is

$$0 = \binom{n}{r+1}.$$

So assume there is an element a not in B . Then $M - a$ has rank r , and has therefore, by induction, at most

$$\binom{n-1}{r+1}$$

circuits.

As all bases of $M - a$ have cardinality r , there are at most

$$\binom{n-1}{r}$$

bases of $M - a$. For each such basis B , $B \cup \{a\}$ has precisely one circuit (a so-called fundamental circuit). Every circuit containing a is the fundamental circuit for some basis. Hence M has at most

$$\binom{n-1}{r}$$

circuits containing a . Therefore the total number of circuits is at most

$$\binom{n-1}{r+1} + \binom{n-1}{r} = \binom{n}{r+1}.$$

□

It is not difficult to see that the inequality in Theorem 1 is an equality if and only if the matroid M is uniform, that is all sets of cardinality r are maximal independent.

Corollary 6.1.3. *For all $1 \leq k \leq n$ we have $W(n, k) \leq \binom{n}{k-1}$.*

Proof. The minimal codewords of an $[n, k]$ code $\mathcal{C}$ are exactly the circuits of the matroid on n elements and rank $n - k$ associated with its dual code. $\square$

We refer to this bound as the *matroid bound*. This bound is tight for $k = 1, n$, and also for $k = n - 1$ as the dual of the repetition code shows.

We can also derive Corollary 6.1.3 from the following:

Theorem 6.1.4. *For all $1 \leq k \leq n$ we have*

$$W(n, k) \leq W(n - 1, k - 1) + \binom{n - 1}{k - 1}.$$

Proof. Let H be the parity check of $\mathcal{C}$ that realizes $W(n, k)$, and H' the matrix with column n removed. Assume, up to column reordering, that there is a basis of the column space not containing column n , or equivalently that the rank of H' is $n - k$. This is always possible if $k \geq 1$. Let $\mathbf{x}$ be a nonzero minimal codeword in $\mathcal{C}$ and discuss according to the value of x_n .

If $x_n = 0$ then the projection $\mathbf{x}'$ on the first $n - 1$ coordinates is a minimal codeword in $\ker(H') = \{\mathbf{y} \in \mathbb{F}_2^{n-1} : H'\mathbf{y}^T = \mathbf{0}^T\}$, an $[n - 1, k - 1]$ code. Therefore, there are at most $W(n - 1, k - 1)$ such vectors.

If $x_n = 1$ then the set of columns where the projection $\mathbf{x}'$ on the first $n - 1$ coordinates is nonzero form an independent set of the column space of H' , because of the minimality property of $\mathbf{x}$. There are at most

$$\binom{n - 1}{n - k} = \binom{n - 1}{k - 1}$$

possible such $\mathbf{x}'$. $\square$

Note that Corollary 6.1.3 also follows from Theorem 6.1.4 by induction on k . Clearly, the bound is true for $k = 1$, since $W(n, 1) = 1 = \binom{n}{0}$. Assuming $W(n-1, k-1) \leq \binom{n-1}{k-2}$, the induction step is completed using Pascal's triangle.

Finally we state, as Theorem 6.1.5, a bound for $W(n, k)$ from [2]. We refer to this bound as the *Agrell bound*.

Theorem 6.1.5. *For $\frac{k-1}{n} > \frac{1}{2}$ we have*

$$W(n, k) \leq \frac{2^k}{4n \binom{\frac{k-1}{n} - \frac{1}{2}}{2}}.$$

6.1.2 Lower bounds

A random coding lower bound was proved in [5], namely

$$W(n, k)2^{n-k} \geq \sum_{j=0}^{n-k+1} \binom{n}{j} \prod_{i=0}^{j-2} (1 - 2^{-(n-k-i)}).$$

Another lower bound is as follows. Denote by $d(n, k)$ the largest minimum distance of an $[n, k]$ code. The following proposition is a direct consequence of [6].

Proposition 6.1.6. *For all $n \geq k \geq 1$, we have*

$$\sum_{i=1}^{\lfloor n/d(n,k) \rfloor} \binom{W(n, k)}{i} \geq 2^k - 1.$$

Proof. Let $\mathcal{C}$ be an $[n, k, d]$ -code. By induction it can be seen that every nonzero codeword can be written as a sum of at most $\lfloor n/d \rfloor$ minimal codewords with pairwise disjoint supports. Hence, enumeration of such sums yields

$$\sum_{i=1}^{\lfloor n/d \rfloor} \binom{W(\mathcal{C})}{i} \geq 2^k - 1.$$

The result follows by choosing $\mathcal{C}$ such that $d = d(n, k)$. □

This result shows that good codes cannot have too few minimum codewords. It is not very sharp. We only get $W(8, 4) \geq 5$, whereas the example of the extended Hamming code (an $[8, 4, 4]$ -code) shows that $W(8, 4) \geq 14$.

6.2 Tabulating $W(n, k)$

6.2.1 Monotonicity properties

It is not difficult to show that $W(n, k) \leq W(n + 1, k)$ by adding a zero column to a code realizing $W(n, k)$. This trivial observation provides better bounds than the random coding bound for $k = 4$ and $n = 7, 8$, as well as for $k = 5$ and $n = 7, 8, 9$ and so on. It is not the case that $W(n, k) \leq W(n, k + 1)$ as the values of $W(4, k)$ already show. We conjecture, but cannot prove, that $W(n, k)$ is an unimodal function of k for fixed n .

Proposition 6.2.1. *For binary codes $\mathcal{C}, \mathcal{D}$ we have $W(\mathcal{C} \oplus \mathcal{D}) = W(\mathcal{C}) + W(\mathcal{D})$.*

Proof. If $\mathbf{u}$ and $\mathbf{v}$ are minimal codewords of $\mathcal{C}$ and $\mathcal{D}$, respectively, then $(\mathbf{u}, 0)$ and $(0, \mathbf{v})$ are minimal codewords of $\mathcal{C} \oplus \mathcal{D}$. Conversely, we claim that all minimal codewords of the latter code arise in that way. Indeed, if $(\mathbf{u}, \mathbf{v})$ is a minimal codeword of $\mathcal{C} \oplus \mathcal{D}$, with both $\mathbf{u}$ and $\mathbf{v}$ nonzero, then $(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, 0) + (0, \mathbf{v})$ contradicting the minimality. □

Using Proposition 2, we see that $W(n, k)$ is superadditive:

$$W(n + m, k + j) \geq W(n, k) + W(m, j).$$

6.2.2 Exact values

Trivial values are $W(n, 1) = 1$, and $W(n, n) = n$ for all $n \geq 1$. $W(n, n-1)$ is already known from Corollary 6.1.3 but a direct elementary proof is tedious.

Proposition 6.2.2. $W(n, n-1) = \binom{n}{2}$ for $3 \leq n$.

Proof. We claim that

$$W(n, n-1) = \max \left\{ \binom{x}{2} + (n-x) : 1 \leq x \leq n \right\}.$$

Denote for $x \geq 2, y \geq 1$ by P_x the $[x, x-1]$ parity-check code, and by U_y the $[y, y]$ universe code. By convention we let $P_1 = \{0\}$, the zero code. It is immediate that $W(P_x) = \binom{x}{2}$, with the convention $\binom{1}{2} = 0$. Also $W(U_y) = y$ because minimal codewords have weight one. We see that, by Proposition 6.2.1, $W(P_x \oplus U_{n-x}) = \binom{x}{2} + (n-x) =: f(x)$. The quadratic function $f(x)$ is increasing for $x \in [1, n]$. Since $f(n) = \binom{n}{2}$, we are done. $\square$

An exact value for $W(n, n-2)$ is possible for n a multiple of 3.

Theorem 6.2.3. *If $\mathcal{C}$ is an $[n, n-2, 2]$ code, then*

$$W(\mathcal{C}) \leq \frac{n^3}{27} + 3 \binom{\frac{n}{3}}{2},$$

with equality if n is a multiple of 3.

Proof. Let $\mathcal{C}$ be an $[n, n-2, 2]$ code. Since $2 \times 2 > n - (n-2) + 1$, the minimal codewords are by [2] those of weight 2 or 3. Let $a+b, a+c, b+c$ be the nonzero dual weights with $a+b+c = n$. Direct counting shows that the number of codewords of weight 3 is abc , and the number of codewords of weight 2 is $\binom{a}{2} + \binom{b}{2} + \binom{c}{2}$. Let

$$ub(a, b, c) = abc + \frac{a^2 - a + b^2 - b + c^2 - c}{2}.$$

The function ub being a symmetric function of the three *real* variables $a, b, c \geq 0$ is maximized under the constraint $a + b + c = n$, when $a = b = c = \frac{n}{3}$. Since $W(\mathcal{C}) \leq ub(a, b, c)$, with a, b, c integers the result follows. $\square$

Corollary 6.2.4. *If $n \geq 3$ then*

$$W(n, n-2) \leq \frac{n^3}{27} + 3 \binom{\frac{n}{3}}{2},$$

with equality if n is a multiple of 3.

Proof. Proof by induction on n . Clearly $W(3, 1) = 1$. If $\mathcal{C}$ is an $[n, n-2, d]$ -code then $d \leq 2$. If $d = 2$ the bound follows by Theorem 6.2.3. If $d = 1$, let $\mathbf{u}$ be a codeword of weight 1. Assume that $u_1 = 1$. Denote by $\mathcal{C}'$ the code shortened at coordinate 1. Thus $\mathcal{C} = \{0, 1\} \oplus \mathcal{C}'$. We know by Proposition 6.2.1 that $W(\mathcal{C}) = 1 + W(\mathcal{C}')$. But $\mathcal{C}'$ is a $[n-1, n-3]$ -code. The bound follows in that case by induction hypothesis.

If n is a multiple of 3 it is easy to construct $\mathcal{C}^\perp$ with single nonzero weight $2n/3$ such that no coordinate is zero ensuring that $\mathcal{C}$ has minimum distance at least 2. Minimum distance 3 is impossible by the Hamming bound. $\square$

6.2.3 Intersecting codes

We recall that a linear binary code meets the trivial bound with equality if and only if it is intersecting. Following [104], denote by $f(k)$ the shortest length of a binary linear intersecting code. Equivalently, there is a function $g(n)$ such that if $k \leq g(n)$ then there is an intersecting $[n, k]$ code; and there is not such a code if $k > g(n)$; thus if $k \leq g(n)$, then $W(n, k) = 2^k - 1$, and if $k > g(n)$, then $W(n, k) \leq 2^k - 2$. The function $g(n)$, the inverse of f , is known exactly for $1 \leq n \leq 15$ [104] and given in Table 6.2.1.

Table 6.2.1: $g(n)$ for $3 \leq n \leq 15$

n	3	4	5	6	7	8	9	10	11	12	13	14	15
$g(n)$	2	2	2	3	3	3	4	4	4	4	5	5	6

6.2.4 Table of $W(n, k)$

The symbols t,m,a in Tables 6.2.2 and 6.2.3 mean that the bounds are obtained as follows.

- t - The trivial bound
- m - The matroid bound
- a - The Agrell bound
- x - The bound of Corollary 6.2.4

When the trivial bound is met with equality the t is omitted. Empty entries correspond to $k > n$ in which case $W(n, k)$ is undefined. The lower bounds are derived by explicit constructions of codes $\mathcal{C}$ derived from the so-called Betten list of indecomposable codes [17], followed by an application of rule G or rule H of [2] to derive $W(\mathcal{C})$. Computations were done using MAGMA [25]. The codes realizing $W(n, k)$ do not in general have the largest minimum distance, but they are in general indecomposable.

6.2.5 Cycle codes of graphs

Let $W_g(n, k)$ be the maximum number of minimal codewords over the set of all $[n, k]$ cycle codes, i.e. it is obtained by restricting $\mathbf{C}[n, k]$ to cycle codes in the definition of

Table 6.2.2: $W(n, k)$ for $3 \leq n \leq 15, 1 \leq k \leq 9$

n/k	1	2	3	4	5	6	7	8	9
3	1	3	3						
4	1	3	6	4					
5	1	3	6	10	5				
6	1	3	7	11	15	6			
7	1	3	7	14	17	21	7		
8	1	3	7	14	22-30 ^t	25	28	8	
9	1	3	7	15	26-30 ^t	33-62 ^t	36	40	9
10	1	3	7	15	30	42-62 ^t	48-126 ^t	48	50
11	1	3	7	15	30	52-62 ^t	66-126 ^t	69-254 ^t	63
12	1	3	7	15	30	54-62 ^t	90-126 ^t	103-254 ^t	95-384 ^a
13	1	3	7	15	31	58-62 ^t	94-126 ^t	151-254 ^t	149-510 ^t
14	1	3	7	15	31	62	106-126 ^t	159-254 ^t	245-510 ^t
15	1	3	7	15	31	63	108-126 ^t	171-254 ^t	245-510 ^t

Table 6.2.3: $W(n, k)$ for $10 \leq n \leq 15, 10 \leq k \leq 15$

n/k	10	11	12	13	14	15
10	10					
11	75	11				
12	82	90	12			
13	130-532 ^a	102-103 ^x	126	13		
14	217-896 ^a	175-796 ^a	126-127 ^x	147	14	
15	385-1022 ^t	308-1228 ^a	221-1253 ^a	155	196	15

$W(n, k)$. The values for $M_g(n, k)$, $1 \leq k \leq 10$, $3 \leq n \leq 15$, are shown in Table 6.2.4. These values were obtained using MAGMA [25] and NAUTY, the graph generation program of B. McKay [86].

6.3 Asymptotic analysis

We consider bounds on $W(n, k)$ for families of binary linear codes of lengths going to infinity and with rates approaching R . We define $\mu(R)$ as the asymptotic exponent of $W(n, k)$ for families of codes $\mathcal{C}$ with rates approaching R , where R is the limsup of k/n . Formally,

$$\mu(R) = \limsup_{n \rightarrow \infty} \frac{1}{n} \log_2 W(n, \lceil Rn \rceil).$$

If, in this definition, we replace $\mathbf{C}[n, k]$ by the cycle codes of graphs with p vertices and q edges (where $n = q$, and $k = q - p + 1$) we obtain $\mu_g(R)$.

By the trivial upper bound, $\mu(R) \leq R$. The matroid bound implies that $\mu(R) \leq$

Table 6.2.4: $W_g(n, k)$ for $1 \leq k \leq 10$, $3 \leq n \leq 15$

n/k	1	2	3	4	5	6	7	8	9	10					
3	1														
4	1														
5	1	3													
6	1	3	7												
7	1	3	7												
8	1	3	7	13											
9	1	3	7	15	22										
10	1	3	7	15	24	37									
11	1	3	7	15	26	39									
12	1	3	7	15	29	42	63								
13	1	3	7	15	29	46	69	91							
14	1	3	7	15	29	51	75	108	133						
15	1	3	7	15	29	57	82	123	156	197					

$H(R)$, where H is Shannon's binary entropy function defined for $x \in [0, 1]$ by

$$H(x) = -x \log_2 x - (1 - x) \log_2 (1 - x).$$

The matroid bound is better than the trivial upper bound for $R > R_0 \approx 0.77$ where $H(R_0) = R_0$.

By averaging arguments (random coding) it was shown in [5] that for $R < 1/2$, we have

$$\mu(R) \geq R,$$

and for $R > 1/2$, we have

$$\mu(R) \geq H(R) - 1 + R.$$

We will prove an upper bound for the number of cycles in a graph with p vertices and q edges. This upper bound will then be used to determine $\mu_g(R)$.

Consider a path with vertices $v_1, v_2, \dots, v_p$. A *multipath* is obtained from a path by replacing some edges with multiple edges. Let $f(q, p)$ denote the maximal number of paths from v_1 to v_p in a multipath with p vertices and q edges. This maximum is attained if no two edge multiplicities differ by more than 1 where the *multiplicity of an edge* is the number of multiple edges sharing the same end vertices.

Lemma 6.3.1. *If x, y are vertices in a graph G with p vertices and q edges, then G has at most $f(q, p)$ paths from x to y .*

Proof. The proof is by induction on p . If $p = 2$ the statement is trivial. So assume $p > 2$. Let d denote the degree of x , and let $x_1, x_2, \dots, x_d$ be the neighbors of x . (Some of these neighbors may be identical.) By induction, $G - x$ has at most $f(q - d, p - 1)$ paths from x_i to y , for each $i = 1, 2, \dots, d$. So G has at most $df(q - d, p - 1)$ paths

from x to y . As $f(q-d, p-1)$ is the number of paths between the ends in a multipath with $p-1$ vertices and $q-d$ edges, we may interpret $df(q-d, p-1)$ as the number of paths between the ends in a multipath with p vertices and q edges, where the first edge multiplicity in the multipath is d . By the maximum property of $f(q, p)$, we have

$$df(q-d, p-1) \leq f(q, p),$$

which completes the proof. $\square$

Theorem 6.3.2. *Let p, q be natural numbers ≥ 2 . There exists a graph with p vertices, q edges, and at least $f(q-1, p)$ cycles. If G is any graph with p vertices and q edges, then G has at most*

$$qf(q-1, p)$$

cycles.

Proof. By the definition of f , there exists a multipath with p vertices, $q-1$ edges, and precisely $f(q-1, p)$ paths between the ends. If we add an edge between the ends we get a graph with p vertices, q edges, and at least $f(q-1, p)$ cycles.

To prove the last statement, consider any edge $e = xy$ in G . The number of cycles in G containing e is the number of paths in $G - e$ from x to y . By Lemma 6.3.1, this number is at most

$$f(q-1, p).$$

This completes the proof. $\square$

Theorem 6.3.3. *For $0 < R \leq 0.5$,*

$$\mu_g(R) = R.$$

If R is of the form $1 - 1/t$ where t is a natural number ≥ 2 , then

$$\mu_g(R) = -(1 - R) \log_2(1 - R).$$

The function $\mu_g(R)$ is continuous and linear in each closed interval from $1 - 1/t$ to $1 - 1/(t + 1)$, where t is a natural number ≥ 2 .

Proof. Assume first that $0 < R < 0.5$. Let tC_p be the cycle of length p where each edge has been duplicated t times. For any two natural numbers p, r , we let $C_{p,r}$ be obtained from $2C_p$ by subdividing one edge r times. Then $C_{p,r}$ has $p + r$ vertices and $n = 2p + r$ edges. Thus the dimension of the cycle code is $k = p + 1$, and the rate of the cycle code is $k/n = (p + 1)/(2p + r)$. For each natural number p we let r be the largest natural number such that $k/n = (p + 1)/(2p + r) \geq R$. Then $k = \lceil Rn \rceil$. Also, $r = \lfloor (p + 1)/R - 2p \rfloor$. Recall that the number of cycles in $2C_p$ and hence also in $C_{p,r}$ is $> 2^p$. Substituting these values in the definition of μ_g and letting p tend to infinity, we conclude that $\mu_g(R) \geq R$.

The trivial upper bound shows that this inequality is, in fact, an equality.

Consider next the case where $R = 1 - 1/t$ for where t is natural number ≥ 2 . The graph tC_p has p vertices and $q = pt$ edges. The number of cycles in this graph is

$$t^p + p \binom{t}{2},$$

the first term counting cycles of length p and the second cycles of length 2. The graph is regular of degree $2t$. Hence it has $n = pt$ edges and has rate $(pt - p + 1)/pt = 1 - 1/t + 1/pt$. Hence its cycle code is an $[n, Rn + 1]$ -code. If we delete an edge, then we get an $[n - 1, \lceil R(n - 1) \rceil]$ -code. Deleting an edge reduces the number of cycles only slightly. Letting p and hence also n tend to infinity, we conclude that

$$\mu_g(R) \geq -(1-R) \log_2(1-R).$$

Now assume that $1 - 1/t < R < 1 - 1/(t+1)$, where t is a (fixed) natural number ≥ 2 . then we let $G(p, R, r)$ denote the graph obtained from tC_p by adding r edges between neighboring vertices such that all edge multiplicities are t or $t+1$. The resulting graph $G(p, R, r)$ has p vertices and $n = pt + r$ edges. Thus the dimension of the cycle code is $k = p(t-1) + r + 1$, and the rate of the cycle code is $k/n = (p(t-1) + r + 1)/(pt + r)$.

We first choose any p so large that the rate of tC_p , which is $(pt - p + 1)/pt = 1 - 1/t + 1/pt$, is smaller than R . Then we let r be the largest natural number such that the rate of $G(p, R, r)$, which is $(p(t-1) + r + 1)/(pt + r)$, is smaller than or equal to R . That is, $r = -pt + \lfloor (p-1)/(1-R) \rfloor$. Then the cycle code of $G(p, R, r)$ is an $[n, \lceil Rn \rceil]$ -code, where $n = pt + r = \lfloor (p-1)/(1-R) \rfloor$ is the number of edges of $G(p, R, r)$.

The number of cycles in $G(p, R, r)$ is $> t^{p-r}(t+1)^r$.

If we substitute these values in the definition of μ_g and let p tend to infinity, then we conclude that

$$\mu_g(R) \geq ((1-R)(1+t) - 1) \log(t) + (-t(1-R) + 1) \log(t+1).$$

The right hand side is clearly a linear function. For R equal to $1 - 1/t$ or $1 - 1/(t+1)$ the right hand side has the same values as the lower bounds we obtained for those two values of R . So we have obtained a lower bound for $\mu_g(R)$ which is continuous and piecewise linear.

We claim that this lower bound is also an upper bound. We used the graph $G(p, R, r)$ above. If we put $q = n = pt + r$, then $G(p, R, r)$ has the maximum number of cycles among those graphs with q edges which are obtained from a cycle of length p by duplicating edges. The graphs used to give the lower bound $f(q-1, p)$ in Theorem 6.3.2 are also graphs of this type. Hence $G(p, R, r)$ has at least $f(q-1, p)$ cycles. On the other hand, Theorem 6.3.2 says that any graph with p vertices and q edges has at most $qf(q-1, p)$ cycles. So no graph with p vertices and q edges has more than q times as many cycles as $G(p, R, r)$. Hence the lower bound for $\mu_g(R)$ obtained from the graphs $G(p, R, r)$ is also an upper bound. $\square$

The function $\mu_g(R)$ is less than the matroid upper bound for all R and also less than the random upper bound for $R > 0.5$. The Agrell upper bound gives the same upper bound on μ as the trivial upper bound. Figure 6.3.1 shows these bounds.

6.4 Conclusion

We investigated in this chapter the maximum number of minimal codewords in binary linear codes. We have given bounds, computed some exact values, and analyzed the asymptotic behavior of the quantity in question.

The computation of the exact values for all n and k is still very much open. A related question is: Can we characterize binary codes attaining the maximum value? Another open problem is the determination of the general asymptotic behavior.

Figure 6.3.1: $\mu(R)$ versus R

Chapter 7

Cheating-immune SSS from codes and cumulative arrays

This chapter focuses on the construction of cheating-immune schemes using binary linear codes and cumulative arrays and is a joint work with Wang Huaxiong [41]. An SSS is cheating-immune if submission of corrupted shares will not give any advantage to the cheaters over the honest participants in the recovery of the original secret.

First, we consider cheating-immune (n, n) -schemes over $\mathbb{F}_2$. The only known constructions of such schemes can be found in [93,94]. By modifying two existing methods to built Boolean functions using binary codes, we were able to produce new schemes. The construction of cheating-immune schemes for other access structures is still an open problem. In line with this, we consider the application of cumulative arrays to obtain cheating-immune (t, n) -schemes.

Section 7.1 gives an overview of secret sharing schemes wherein some participants submit invalid shares during a reconstruction of the secret. We briefly discuss several

types of secret sharing schemes with some form of security against cheating. In Section 7.2, we present the concept and relevant properties of cheating-immune secret sharing schemes. Section 7.3 discusses the relation between cheating-immune schemes and Boolean functions. We present constructions of cheating-immune and strictly cheating-immune schemes using binary linear codes in Sections 7.4 and 7.5. These constructions produce new cheating-immune schemes. In Section 7.6, we study the application of cumulative arrays to build cheating-immune threshold schemes.

7.1 Secret sharing schemes with cheaters

In this chapter, we consider the setting where members of an authorized set submit their shares to an entity called a *combiner* to reconstruct the secret. The combiner computes the secret and transmits it to the members of the authorized set. We assume that the dealer and the combiner are honest, i.e., they always follow the protocol, but participants can cheat by submitting corrupted shares during the reconstruction stage.

7.1.1 Tompa-Woll attack

Tompa and Woll [108] demonstrated that a participant can cheat in a Shamir threshold scheme. We illustrate this with the following example.

Example 7.1.1. *Suppose we have a Shamir (t, n) -threshold scheme and that a secret is shared using the polynomial $f(x)$. Let $\{P_{i_1}, P_{i_2}, \dots, P_{i_t}\}$ be a set of t participants who wants to reconstruct the secret. For $j = 1, 2, \dots, t$, assume that the share of P_{i_j} is $s_{i_j} = f(x_{i_j})$. Suppose that P_{i_1} cheats by submitting a value $s'_{i_1} \neq s_{i_1}$ to the combiner. By using Lagrange interpolation, the combiner will output a polynomial*

$g(x)$ such that $g(x_{i_j}) = f(x_{i_j})$ for $j = 2, 3, \dots, t$, and $g(x_{i_1}) = s'_{i_1}$. Then P_{i_1} can construct the polynomial $h(x) = f(x) - g(x)$ since it knows k points of $h(x)$. Thus, P_{i_1} can compute $f(0) = h(0) + g(0)$.

We can see that not only does the cheater learn the secret, but the honest participants are also prevented from knowing it. This attack actually applies to any linear secret sharing scheme. Tompa and Woll also showed a modification of the Shamir threshold scheme to deal with cheating. Let $\epsilon > 0$ and $\mathbb{F}_q$ be a finite field such that

$$q > n \text{ and } \frac{(q-1)(t-1)}{q-t} < \epsilon.$$

The secrets and shares will be taken from $\mathbb{F}_q$. Suppose the secret is $s \in \mathbb{F}_q$. The dealer chooses $t-1$ elements $a_1, \dots, a_{t-1}$ from $\mathbb{F}_q$ independently at random and with uniform distribution. The next step is to construct the polynomial

$$f(x) = s + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}.$$

The dealer then selects a set $(x_1, x_2, \dots, x_n)$ uniformly and randomly from the set of all n -permutations of distinct nonzero elements of $\mathbb{F}_q$. The share of P_i is given by $s_i = (x_i, f(x_i))$ for $i = 1, \dots, n$. Note that the size of the share is twice the size of the secret.

For this scheme, the probability of successful cheating by $t-1$ participants is at most $(q-1)(t-1)/(q-t)$, where *successful cheating* means that the $t-1$ cheaters were able to learn the secret and the honest participant was prevented from knowing it and was not able to detect that cheating occurred. Hence, we can choose q accordingly so that the probability of successful cheating is less than ϵ .

One weakness of the scheme is that even if cheating is detected, the dishonest participants still learn the secret while the honest ones do not. In order to prevent

the cheaters from knowing the secret, Tompa and Woll proposed a second scheme that is an improvement of the first but at the cost of larger shares.

7.1.2 Security of SSS from linear codes against cheating

The secret sharing schemes from linear codes discussed in Section 3.2, Chapter 3 have some form of security against cheating. Recall that in these schemes, the shares form a codeword of a linear code. Thus, we can use the error-detecting and error-correcting properties of the linear code. This observation was first noted in threshold schemes from MDS codes [67, 85]. The general case was first considered in [83, 96].

The schemes from linear codes have the capability to detect cheating or even identify the invalid shares and recover the correct secret. These capabilities are achieved at the cost of requiring more than the minimum number of shares needed to determine the secret. As first pointed out in [85], suppose that in a (t, n) -threshold scheme there are $s > t$ shares and v of which are invalid. If $s - v \geq t$ then cheating can be detected. If $s - 2v \geq t$ then the invalid shares can be identified and corrected.

For the general case, suppose all participants submit their shares to the combiner. Let d be the minimum distance of the underlying linear code. Then at most $d - 1$ corrupted shares can be detected and at most $\lfloor (d - 1)/2 \rfloor$ corrupted shares can be corrected.

7.1.3 Other SSS secure against cheating

Secret sharing schemes having the property that the correct secret can still be recovered even if some of the submitted shares are invalid are called *robust secret sharing*

schemes. A unified study of robust schemes can be found in [13]. We have seen that the secret sharing schemes from linear codes can be robust. A different kind of robust schemes is given by *verifiable secret sharing schemes* [34, 95]. To achieve robustness, the shares in these schemes contain additional information so that the shares can be checked for correctness. These schemes are actually more general in the sense that it also deals with a dishonest dealer.

Some secret sharing schemes have the capability to detect cheating or identify the cheating participants but are not necessarily robust schemes. Examples can be found in [69, 89, 97, 108]. These schemes also require shares with extra information to be used for cheater detection or identification. This results into an increase in the size of the shares. The paper [80] gives a survey and comparison of the different secret sharing schemes that deal with cheating.

7.2 Cheating-immune secret sharing schemes

Another type of approach to the cheating problem was proposed by Pieprzyk and Zhang in [94] by introducing the concept of *cheating-immune secret sharing scheme*. Informally speaking, a secret sharing scheme is said to be cheating-immune if submission of corrupted shares will not give any advantage to the cheaters over the honest participants in the recovery of the original secret. In a cheating-immune scheme, even if the cheaters use the incorrect secret returned by the combiner together with their own valid shares, they will not gain more information about the correct secret than an honest participant.

The approach in a cheating-immune secret sharing scheme is different from the

schemes mentioned above. When participants cheat and the combiner returns an incorrect secret, the goal is to prevent the cheaters from knowing the correct secret. There is no cheating detection and the honest participants do not learn the correct secret. The advantage of a cheating-immune scheme is that it does not require extra information on the shares or additional shares during the reconstruction stage. The shares can be of the same size as the secret.

We now present a model of cheating-immune secret sharing scheme taken from [92]. The model considers (n, n) -secret sharing scheme where the secrets and shares are elements of the finite field $\mathbb{F}_q$, where q is a power of prime.

Given two vectors $\alpha, \beta \in \mathbb{F}_q^n$, we say that $\alpha \preceq \beta$ if $\alpha_j \neq 0$ implies $\beta_j \neq 0$. The shares of the participants will be represented by a vector in $\mathbb{F}_q^n$. An (n, n) -secret sharing scheme can be represented by a *defining function*

$$f : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$$

which maps to each vector of shares a secret value in $\mathbb{F}_q$.

The cheaters are represented by a vector $\delta \in \mathbb{F}_q^n$, called a *cheating vector*. The nonzero entries of δ represent the modifications done by the cheaters. It follows that $\text{wt}(\delta)$ gives the number of cheaters. Given two vectors x and δ , we define $x_\delta^+ \in \mathbb{F}_q^n$ as the vector with $x_j^+ = x_j$ if $\delta_j \neq 0$, and $x_j^+ = 0$ otherwise. We also define $x_\delta^- \in \mathbb{F}_q^n$ as the vector with $x_j^- = x_j$ if $\delta_j = 0$, and $x_j^- = 0$ otherwise.

Now, consider the following two sets:

$$R(\delta, \alpha_\delta^+, K) = \{x_\delta^- : f(x_\delta^- + \alpha_\delta^+) = K\}$$

and

$$R(\delta, \alpha_\delta^+ + \delta, K^*) = \{x_\delta^- : f(x_\delta^- + \alpha_\delta^+ + \delta) = K^*\}.$$

The first set is the collection of the possible shares of honest participants such that together with the cheaters' valid shares, the original secret K will be reconstructed. The second set is the collection of all the possible shares of honest participants such that together with the cheaters' corrupted shares, the secret K^* will be reconstructed. The value

$$\rho_{\delta, \alpha} = |R(\delta, \alpha_{\delta}^+ + \delta, K^*) \cap R(\delta, \alpha_{\delta}^+, K)| / |R(\delta, \alpha_{\delta}^+ + \delta, K^*)|$$

is the *probability of successful cheating with respect to δ, α* .

Definition 7.2.1. An (n, n) -secret sharing scheme over $\mathbb{F}_q$ is said to be *k-cheating immune* if, for every $\alpha \in \mathbb{F}_q^n$ and any $\delta \in \mathbb{F}_q^n$, with $1 \leq \text{wt}(\delta) \leq k$, it holds that $\rho_{\delta, \alpha} = 1/q$.

We will use the notation $k\text{-CI } (n, n)\text{-SSS}$ to denote an (n, n) -secret sharing scheme that is k -cheating-immune. The integer k is called the *order of cheating immunity*. Note that in the definition, we assumed that all cheaters submit fake shares. There is a more general type of cheating-immune secret sharing scheme in which the case when not all cheaters submit fake shares is considered. For this type of scheme, called *strictly cheating-immune secret sharing scheme*, we use two vectors $\delta \in \mathbb{F}_2^n, \tau \in \mathbb{F}_q^n$. The vector δ represents all cheaters while the vector τ represents the cheaters who submitted corrupted shares. Thus, we have $\tau \preceq \delta$. We define

$$\rho_{\delta, \tau, \alpha} = |R(\delta, \alpha_{\delta}^+ + \tau, K^*) \cap R(\delta, \alpha_{\delta}^+, K)| / |R(\delta, \alpha_{\delta}^+ + \tau, K^*)|$$

as the *probability of successful cheating with respect to δ, τ, α* .

Definition 7.2.2. An (n, n) -secret sharing scheme over $\mathbb{F}_q$ is said to be *strictly k-cheating immune* if, for every $\alpha \in \mathbb{F}_q^n$, any $\delta \in \mathbb{F}_2^n$ and any $\tau \in \mathbb{F}_q^n$, with $\tau \preceq \delta$,

$1 \leq \text{wt}(\boldsymbol{\tau}) \leq \text{wt}(\boldsymbol{\delta}) \leq k$, it holds that $\rho_{\boldsymbol{\delta}, \boldsymbol{\tau}, \boldsymbol{\alpha}} = 1/q$.

Constructions of cheating-immune secret sharing schemes can be found in [39, 75, 76, 92–94]. It was shown in [39] that cheating-immune secret sharing schemes cannot be perfect. We also have the following bounds on the number of cheaters.

Theorem 7.2.3. [39] *An (n, n) -secret sharing scheme over $\mathbb{F}_q$ with defining function f can be k -cheating-immune only if $2k < n$.*

Theorem 7.2.4. [26] *An (n, n) -secret sharing scheme over $\mathbb{F}_2$ with defining function f can be k -cheating-immune only if $2k \leq n - 2$.*

Example 7.2.5. *Here is an example of a 1-CI $(4, 4)$ -SSS taken from [94]. The defining function is given by $f(x_1, x_2, x_3, x_4) = x_1x_2 \oplus x_1x_4 \oplus x_2x_3 \oplus x_3x_4 \oplus x_3 \oplus x_4$. Table 7.2.1 shows the possible distribution of shares.*

Table 7.2.1: Share distribution of a 1-CI $(4, 4)$ -SSS

f	0	1	1	1	0	1	0	0	0	0	1	0	1	1	1	0
P_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
P_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
P_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
P_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

7.3 Cheating-immune schemes and Boolean functions

The defining function of an (n, n) -secret sharing scheme over $\mathbb{F}_2$ is a Boolean function. In this section, we review the relation between cheating-immune schemes and Boolean functions.

Let f be a function from $\mathbb{F}_2^n \rightarrow \mathbb{F}_2$. We will also use the notation $f(\mathbf{x})$ where $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_2^n$. We say that f is *balanced* if the number of vectors $\mathbf{x}$ that is mapped to 0 is equal to the number of vectors that is mapped to 1 (this number is 2^{n-1}).

An *affine* function is of the form $f(x_1, \dots, x_n) = a_1x_1 \oplus \dots \oplus a_nx_n \oplus c$. When $c = 0$ then f is a *linear* function. Any non-constant affine function is balanced.

We say that f satisfies the *propagation criterion of degree l* or $\text{PC}(l)$ if $f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \boldsymbol{\alpha})$ is balanced for any vector $\boldsymbol{\alpha} \in \mathbb{F}_2^n$ such that $1 \leq \text{wt}(\boldsymbol{\alpha}) \leq l$. When $l = 1$, $\text{PC}(1)$ is also known as the *strict avalanche criterion* or SAC.

A Boolean function f is said to be t -resilient if for every subset $\{j_1, \dots, j_t\} \subseteq \{1, \dots, n\}$ and every $(a_1, \dots, a_t) \in \mathbb{F}_2^t$, the function

$$f(x_1, \dots, x_n)|_{x_{j_1}=a_1, \dots, x_{j_t}=a_t}$$

is balanced over $\mathbb{F}_2^{n-t}$. If f is t -resilient then it is also s -resilient for $0 \leq s \leq t$.

The *Walsh transform* of f is a real-valued function on $\mathbb{F}_2^n$ defined as

$$\hat{f}_\chi(\mathbf{u}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{u} \cdot \mathbf{x}}$$

where $\mathbf{u} \cdot \mathbf{x}$ denotes the usual inner product. A Boolean function f is t -resilient if and only if $\hat{f}_\chi(\mathbf{u}) = 0$ for all $\mathbf{u} \in \mathbb{F}_2^n$ such that $\text{wt}(\mathbf{u}) \leq t$ [55].

The following theorem is a characterization of the defining function of an (n, n) -scheme that is 1-cheating-immune:

Theorem 7.3.1. *[93] An (n, n) -secret sharing scheme with defining function f is 1-cheating-immune if and only if f is 1-resilient and satisfies the SAC.*

To characterize k -cheating-immune schemes with $k > 1$, Pieprzyk and Zhang introduced the concept of strengthened propagation in [93]. We say that a function f satisfies the *strengthened propagation of degree l* or $SP(l)$ if for any $\delta \in \mathbb{F}_2^n$, with $1 \leq \text{wt}(\delta) \leq l$, and for any $\tau \preceq \delta$, the function $f(x_\delta^- \oplus \tau) \oplus f(x_\delta^- \oplus \tau \oplus \delta)$ is balanced.

Theorem 7.3.2. *[93] An (n, n) -secret sharing scheme with defining function f is k -cheating-immune if and only if f is k -resilient and satisfies the $SP(k)$.*

The next theorem is a characterization of strictly k -cheating-immune schemes.

Theorem 7.3.3. *[93] An (n, n) -secret sharing scheme with defining function f is strictly k -cheating-immune if and only if the following conditions are satisfied:*

- i. f is k -resilient.*
- ii. For any integer t with $0 \leq t \leq k - 1$, every function obtained from f by keeping any t input bits constant satisfies the $SP(k - t)$.*

7.4 Cheating-immune schemes from binary linear codes

We revisit a construction of resilient functions satisfying the propagation criterion presented in [112]. This construction is a special case of a more general construction

that can be found in [29]. By adding a condition, we prove that the resilient functions obtained also satisfy the strengthened propagation. Thus, these functions can be used to construct cheating-immune secret sharing schemes.

7.4.1 Construction of cheating-immune schemes

Theorem 7.4.1. *Let $\mathcal{C}$ be a binary $[n - m, m, d]$ linear code with dual distance $d^\perp$. Assume that $d^\perp \leq d$. Let G be a generator matrix of $\mathcal{C}$ and let α be a vector in $\mathbb{F}_2^{n-m}$ which is not in $\mathcal{C}$. Consider the function*

$$f(\mathbf{x}, \mathbf{y}) = \mathbf{x}G\mathbf{y}^T \oplus \alpha\mathbf{y}^T,$$

where $\mathbf{x} \in \mathbb{F}_2^m, \mathbf{y} \in \mathbb{F}_2^{n-m}$ and $\oplus$ is the addition over $\mathbb{F}_2$. Then we have the following:

- i. f is t -resilient with $t = \min_{\mathbf{c} \in \mathcal{C}} \{\text{wt}(\mathbf{c} \oplus \alpha)\} - 1$.
- ii. f satisfies the strengthened propagation of degree l with $l = d^\perp - 1$.

Proof. We will use the Walsh transform of f to prove that it is t -resilient. For $\mathbf{a} \in \mathbb{F}_2^m, \mathbf{b} \in \mathbb{F}_2^{n-m}$, we have

$$\begin{aligned} \hat{f}_\chi(\mathbf{a}, \mathbf{b}) &= \sum_{(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}, \mathbf{y}) \oplus (\mathbf{a}, \mathbf{b}) \cdot (\mathbf{x}, \mathbf{y})} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\mathbf{a} \cdot \mathbf{x}} \sum_{\mathbf{y} \in \mathbb{F}_2^{n-m}} (-1)^{(\mathbf{x}G \oplus \alpha \oplus \mathbf{b}) \cdot \mathbf{y}}. \end{aligned}$$

Since $(\mathbf{x}G \oplus \alpha \oplus \mathbf{b}) \cdot \mathbf{y}$ is a linear function of $\mathbf{y}$ (for a given $\mathbf{x}$) then it is either balanced or zero. It follows that

$$\hat{f}_\chi(\mathbf{a}, \mathbf{b}) = 2^{n-m} (-1)^{\mathbf{a} \cdot \mathbf{x}'}$$

where $\mathbf{x}'$ is the solution to the equation $\mathbf{x}G \oplus \alpha \oplus \mathbf{b} = 0$. Let $t = \min_{\mathbf{c} \in \mathcal{C}} \{\text{wt}(\mathbf{c} \oplus \alpha)\} - 1$.

If $\text{wt}(\mathbf{b}) \leq t$ then $\hat{f}_\chi(\mathbf{a}, \mathbf{b}) = 0$. Thus, f is t -resilient.

Now, let $l = d^\perp - 1$. Let $\delta, \tau \in \mathbb{F}_2^n$ such that $1 \leq \text{wt}(\delta) \leq l$ and $\tau \preceq \delta$. Suppose $z = (x, y)$ and $\delta = (\delta^x, \delta^y)$ where $\delta^x \in \mathbb{F}_2^m, \delta^y \in \mathbb{F}_2^{n-m}$. Similarly, we will write $\tau = (\tau^x, \tau^y)$. We have

$$\begin{aligned}
& f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta) \\
&= (x_{\delta^x}^- \oplus \tau^x)G(y_{\delta^y}^- \oplus \tau^y)^T \oplus \alpha(y_{\delta^y}^- \oplus \tau^y)^T \\
&\oplus (x_{\delta^x}^- \oplus \tau^x \oplus \delta^x)G(y_{\delta^y}^- \oplus \tau^y \oplus \delta^y)^T \oplus \alpha(y_{\delta^y}^- \oplus \tau^y \oplus \delta^y)^T \\
&= (x_{\delta^x}^- \oplus \tau^x)G(\delta^y)^T \oplus \delta^x G(y_{\delta^y}^- \oplus \tau^y)^T \oplus \delta^x G(\delta^y)^T \oplus \alpha(\delta^y)^T.
\end{aligned}$$

We consider three cases where we will prove that $f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta)$ is balanced by showing that it is a non-constant affine function. First, suppose $\delta^y = 0$. Then $f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta) = \delta^x G y^T$. Since $\delta^x \neq 0$ then $\delta^x G$ is a nonzero codeword of $\mathcal{C}$. Hence, $\delta^x G y^T$ is a non-constant affine function.

Next, suppose that $\delta^x = 0$. Then $f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta) = xG(\delta^y)^T \oplus \alpha(\delta^y)^T$. Since $0 < \text{wt}(\delta^y) < d^\perp$ then $G(\delta^y)^T \neq 0$. Thus, $xG(\delta^y)^T \oplus \alpha(\delta^y)^T$ is a non-constant affine function.

Finally, suppose $\delta^x \neq 0$ and $\delta^y \neq 0$. We examine the term $\delta^x G(y_{\delta^y}^- \oplus \tau^y)^T$ in the expression for $f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta)$ above. Since $\delta^x \neq 0$ then $\delta^x G$ is a nonzero codeword of $\mathcal{C}$. The number of constant entries of $(y_{\delta^y}^- \oplus \tau^y)$ is equal to $\text{wt}(\delta^y)$. It follows that $\delta^x G(y_{\delta^y}^- \oplus \tau^y)^T$ is a non-constant affine function since $\text{wt}(\delta^y) < d^\perp \leq d \leq \text{wt}(\delta^x G)$. This means that $f(z_\delta^- \oplus \tau) \oplus f(z_\delta^- \oplus \tau \oplus \delta)$ is a non-constant affine function. Therefore, f satisfies the strengthened propagation of degree l . $\square$

By combining the preceding theorem with Theorem 7.3.2, we have the following:

Corollary 7.4.2. *Let $\mathcal{C}$ be a binary $[n - m, m, d]$ linear code with dual distance $d^\perp$ and let α be a vector in $\mathbb{F}_2^{n-m}$ which is not in $\mathcal{C}$. Assume that $d^\perp \leq d$. Let $l = d^\perp - 1$ and $t = \min_{\mathbf{c} \in \mathcal{C}} \{\text{wt}(\mathbf{c} \oplus \alpha)\} - 1$. Then there exists a k -CI (n, n) -SSS where $k = \min\{t, l\}$.*

We remark that given $\mathcal{C}$, we can maximize the value of $t = \min_{\mathbf{c} \in \mathcal{C}} \{\text{wt}(\mathbf{c} \oplus \alpha)\} - 1$ by choosing α such that $\min_{\mathbf{c} \in \mathcal{C}} \{\text{wt}(\mathbf{c} \oplus \alpha)\}$ is equal to the *covering radius* $\rho(\mathcal{C})$ of $\mathcal{C}$ (see [59]).

7.4.2 Existence of cheating-immune schemes

There are two known methods to construct cheating-immune secret sharing schemes over $\mathbb{F}_2$ due to Pieprzyk and Zhang. The first one [94] uses bent functions to build 1-CI (n, n) -SSS for n even. Given positive integers k and s with $s \geq k + 1$, the second method [93] allows us to build a k -CI (n, n) -SSS where $n = n_1 + \dots + n_s$, $n_i = 2k + 1$ or $2k + 2$, $i = 1, \dots, s$. If $n \geq (2k + 1)^2$ then there exists a k -CI (n, n) -SSS. It is also possible to use the second method to construct k -CI (n, n) -SSS for some values of n such that $n \geq (2k + 1)(k + 1)$. We now apply the construction described in the previous subsection to obtain cheating-immune schemes that were not shown to exist before (we only show the case for $k = 1, 2, 3$). We used MAGMA [25] in searching for suitable linear codes.

1-CI (n, n) -SSS. We will show the existence of a 1-CI $(5, 5)$ -SSS. Let $\mathcal{C}$ be the binary $[4, 1, 4]$ linear code. Then we have $G = [1111]$ and $d^\perp = 2$. Let $\alpha = (1, 1, 0, 0)$. Then $t = 1$. Applying Theorem 7.4.1, we obtain $f(\mathbf{x}, \mathbf{y}) = x(y_1 + y_2 + y_3 + y_4) + y_1 + y_2$. We then have a 1-CI $(5, 5)$ -SSS with defining function given by f .

In general, given a binary $[n', 1, n']$ linear code $\mathcal{C}$, $n' \geq 4$, then $d^\perp = 2$ and its

covering radius $\rho(\mathcal{C}) \geq 2$. Hence, we can obtain an $(n' + 1, n' + 1)$ -secret sharing scheme that is 1-cheating-immune for $n' \geq 4$.

2-CI (n, n) -SSS. In this case, we obtained schemes for $n = 10, 13, 14, 19$. Consider the binary simplex code $S_{m,2}$ with parameters $[2^m - 1, m, 2^{m-1}]$. Then $d^\perp = 3$ and the covering radius is equal to $\rho = 2^{m-1} - 1$ (see [59]). For $m \geq 3$, we can obtain a $(2^m - 1 + m, 2^m - 1 + m)$ -secret sharing scheme that is 2-cheating-immune. When $m = 3$, we have a 2-CI $(10, 10)$ -SSS. When $m = 4$, we have a 2-CI $(19, 19)$ -SSS.

We found two linear codes with $d^\perp = 3$ and $\rho = 3$. The first one is a $[9, 4, 3]$ code that leads to a 2-CI $(13, 13)$ -SSS. The second one is a $[10, 4, 3]$ code that leads to a 2-CI $(14, 14)$ -SSS.

3-CI (n, n) -SSS. We were able to search for codes that can be used to construct 3-CI (n, n) -SSS for $n = 18-27, 33, 34, 41$.

We note that by Theorem 7.2.4, $n \geq 2k + 2$. For $k = 1$, the existence problem is complete ($n = 5$ is the only open case and we gave an example above). For $k = 2$, it is still open for $n = 6-9, 11, 12$. For $k = 3$, it is still open for $n = 8-17$.

7.5 Strictly cheating-immune schemes from binary linear codes

We consider a generalization of the construction in Theorem 7.4.1 which will enable us to build strictly cheating-immune secret sharing schemes. The resulting construction is a special case of the one given by Kurosawa and Satoh in [71] to design Boolean functions satisfying higher order $\text{PC}(l)$.

7.5.1 Construction of strictly cheating-immune schemes

Theorem 7.5.1. *Let Q be an $m \times (n - m)$ matrix over $\mathbb{F}_2$ and l be a positive integer such that the following conditions are satisfied:*

- i. $m > l$ and $n - m > l$*
- ii. For $h = 0, \dots, l - 1$, $\text{wt}(Q\mathbf{r}^T) \geq l$ for every $\mathbf{r} \in \mathbb{F}_2^{n-m}$ such that $1 \leq \text{wt}(\mathbf{r}) \leq l - h$.*
- iii. For $h = 0, \dots, l - 1$, $\text{wt}(\mathbf{s}Q) \geq l$ for every $\mathbf{s} \in \mathbb{F}_2^m$ such that $1 \leq \text{wt}(\mathbf{s}) \leq l - h$.*

Define the function $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}Q\mathbf{y}^T \oplus \boldsymbol{\alpha}\mathbf{y}^T$ where $\mathbf{x} \in \mathbb{F}_2^m$, $\mathbf{y}, \boldsymbol{\alpha} \in \mathbb{F}_2^{n-m}$ and $\boldsymbol{\alpha} \notin \{\mathbf{s}Q : \mathbf{s} \in \mathbb{F}_2^m\}$. Then we have the following properties of f :

- 1. f is t -resilient with $t = \min_{\mathbf{s} \in \mathbb{F}_2^m} \{\text{wt}(\mathbf{s}Q \oplus \boldsymbol{\alpha})\} - 1$.*
- 2. When we keep any h coordinates of the input constant, then f satisfies the strengthened propagation of degree $l - h$.*

Proof. Consider the Walsh transform of f . For $\mathbf{a} \in \mathbb{F}_2^m$, $\mathbf{b} \in \mathbb{F}_2^{n-m}$, we have

$$\begin{aligned} \hat{f}_\chi(\mathbf{a}, \mathbf{b}) &= \sum_{(\mathbf{x}, \mathbf{y}) \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}, \mathbf{y}) \oplus (\mathbf{a}, \mathbf{b}) \cdot (\mathbf{x}, \mathbf{y})} \\ &= \sum_{\mathbf{x} \in \mathbb{F}_2^m} (-1)^{\mathbf{a} \cdot \mathbf{x}} \sum_{\mathbf{y} \in \mathbb{F}_2^{n-m}} (-1)^{(\mathbf{x}Q \oplus \boldsymbol{\alpha} \oplus \mathbf{b}) \cdot \mathbf{y}}. \end{aligned}$$

For every $\mathbf{x}$, $(\mathbf{x}Q \oplus \boldsymbol{\alpha} \oplus \mathbf{b}) \cdot \mathbf{y}$ is a linear function in $\mathbf{y}$. Hence, it is either balanced or zero. It follows that

$$\hat{f}_\chi(\mathbf{a}, \mathbf{b}) = 2^{n-m} \sum_{\mathbf{x} \in S} (-1)^{\mathbf{a} \cdot \mathbf{x}},$$

where S is the set of solutions to the equation $\mathbf{x}Q \oplus \boldsymbol{\alpha} \oplus \mathbf{b} = \mathbf{0}$. Let $t = \min_{\mathbf{s} \in \mathbb{F}_2^m} \{\text{wt}(\mathbf{s}Q \oplus \boldsymbol{\alpha})\} - 1$. If $\text{wt}(\mathbf{b}) \leq t$ then $\hat{f}_\chi(\mathbf{a}, \mathbf{b}) = 0$. Thus, f is t -resilient.

Now we prove (2). Suppose we keep any h coordinates of the input constant. Without loss of generality, we assume that $x_1 = a_1, \dots, x_u = a_u$ and $y_1 = b_1, \dots, y_v = b_v$, where $u + v = h, u < m, v < n - m$. Let $\underline{x} = (a_1, \dots, a_u, x_{u+1}, \dots, x_m)$, $\underline{y} = (b_1, \dots, b_v, y_{v+1}, \dots, y_{n-m})$ and $\underline{z} = (\underline{x}, \underline{y})$. We need to show that $f(\underline{z})$ satisfies strengthened propagation of degree $l - h$ (as a function on $\mathbb{F}_2^{n-h}$). In particular, we need to prove that the function $f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta)$ is balanced for any $\delta, \tau \in \mathbb{F}_2^n$ such that $\tau \preceq \delta$, $1 \leq \text{wt}(\delta) \leq l - h$ and the set of nonzero coordinates of δ is a subset of $\{u + 1, \dots, m\} \cup \{v + 1, \dots, n - m\}$. We use the notation $\delta = (\delta^x, \delta^y)$ to distinguish between the coordinates corresponding to $\underline{x}$ and $\underline{y}$.

The function $f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta)$ is equal to

$$(\underline{x}_{\delta^x}^- \oplus \tau^x)Q(\delta^y)^T \oplus \delta^x Q(\underline{y}_{\delta^y}^- \oplus \tau^y)^T \oplus \delta^x Q(\delta^y)^T \oplus \alpha(\delta^y)^T.$$

Suppose that $\text{wt}(\delta^x) = 0$. Then

$$f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta) = \underline{x}Q(\delta^y)^T \oplus \alpha(\delta^y)^T.$$

Since $u < h < l$ and $\text{wt}(Q(\delta^y)^T) \geq l$ then $f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta)$ is a non-constant affine function.

Suppose that $\text{wt}(\delta^y) = 0$. Then

$$f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta) = \delta^x Q \underline{y}^T.$$

Since $v < h < l$ and $\text{wt}(\delta^x Q) \geq l$ then $f(\underline{z}_\delta^- \oplus \tau) \oplus f(\underline{z}_\delta^- \oplus \tau \oplus \delta)$ is a non-constant affine function.

Suppose that $\text{wt}(\delta^x) \neq 0$ and $\text{wt}(\delta^y) \neq 0$. Examine the term $(\underline{x}_{\delta^x}^- \oplus \tau^x)Q(\delta^y)^T$.

The number of constant entries of the vector $\underline{x}_{\delta^x}^- \oplus \tau^x$ is equal to $u + \text{wt}(\delta^x) <$

$h + (l - h) = l$. Since $\text{wt}(Q(\delta^\mathbf{y})^T) \geq l$ then it follows that $f(\mathbf{z}_\delta^- \oplus \boldsymbol{\tau}) \oplus f(\mathbf{z}_\delta^- \oplus \boldsymbol{\tau} \oplus \delta)$ is a non-constant affine function. $\square$

By combining Theorem 7.5.1 with Theorem 7.3.3, we obtain the following result.

Corollary 7.5.2. *Let Q be an $m \times (n - m)$ matrix over $\mathbb{F}_2$ and l be a positive integer such that the conditions in Theorem 7.5.1 are satisfied. Then there exists a strictly k -CI (n, n) -SSS where $k = \min\{t, l\}$.*

The following theorem is a special case of Theorem 19 in [71].

Theorem 7.5.3. *Let $\mathcal{C}_1$ be a binary $[m, w, d_1]$ linear code and $\mathcal{C}_2$ be a binary $[n - m, w, d_2]$ linear code such that $\min\{d_1^\perp, d_2^\perp\} - 1 \leq \min\{d_1, d_2\}$. Let G_i be a generator matrix of $\mathcal{C}_i$, $i = 1, 2$. Define $Q = G_1^T G_2$ and $l = \min\{d_1^\perp, d_2^\perp\} - 1$. Then we have the following:*

- i. *For $h = 0, \dots, l - 1$, $\text{wt}(Q\mathbf{r}^T) \geq l$ for every $\mathbf{r} \in \mathbb{F}_2^{n-m}$ such that $1 \leq \text{wt}(\mathbf{r}) \leq l - h$.*
- ii. *For $h = 0, \dots, l - 1$, $\text{wt}(\mathbf{s}Q) \geq l$ for every $\mathbf{s} \in \mathbb{F}_2^m$ such that $1 \leq \text{wt}(\mathbf{s}) \leq l - h$.*

If we use Theorem 7.5.3 to construct strictly cheating-immune schemes then we need to determine the parameters t and l as defined in Theorem 7.5.1. We recall that $t = \min_{\mathbf{s} \in \mathbb{F}_2^m} \{\text{wt}(\mathbf{s}Q \oplus \boldsymbol{\alpha})\} - 1$ where $\boldsymbol{\alpha} \notin \{\mathbf{s}Q : \mathbf{s} \in \mathbb{F}_2^m\}$. If $Q = G_1^T G_2$ then $\{\mathbf{s}Q : \mathbf{s} \in \mathbb{F}_2^m\}$ is a subcode of $\mathcal{C}_2$. It follows that to maximize t , choose $\boldsymbol{\alpha}$ such that $\min_{\mathbf{s} \in \mathbb{F}_2^m} \{\text{wt}(\mathbf{s}Q \oplus \boldsymbol{\alpha})\}$ is equal to the covering radius of the subcode.

7.5.2 Existence of strictly cheating-immune schemes

The only known method to build strictly cheating-immune secret sharing schemes over $\mathbb{F}_2$ is given by Pieprzyk and Zhang in [93]. Here we give some schemes (by applying the method above) which cannot be obtained by their construction.

Strictly 2-CI (n, n) -SSS. For $k = 2$, the construction in [93] gives schemes for $n = n_1 + \dots + n_s$ where $s \geq 3$ and $n_i = 5$ or 6 , $i = 1, \dots, s$. We show the existence of schemes for $n = 14, 19$.

Consider again the binary simplex code $S_{m,2}$ with parameters $[2^m - 1, m, 2^{m-1}]$ and $d^\perp = 3$. Let $Q = G^T G$ where G is a generator matrix of S_m . Then $l = 2$. For any subcode $\mathcal{C}' \subseteq S_m$, the covering radius $\rho(\mathcal{C}') \geq 2^{m-1}$ (see [59, Lemma 11.1.5]). For $m \geq 3$, it is possible to construct a strictly 2-CI $(2^{m+1} - 2, 2^{m+1} - 2)$ -SSS. When $m = 3$ then we have $n = 14$.

To construct a strictly 2-CI $(19, 19)$ -SSS, we use the linear codes with parameters $[9, 4, 3]$ and $[10, 4, 3]$ ($d^\perp = 3$, $\rho = 3$) given in Section 3.2.

Strictly 3-CI (n, n) -SSS. For $k = 3$, the construction in [93] gives schemes for $n = n_1 + \dots + n_s$ where $s \geq 4$ and $n_i = 9$ or 10 , $i = 1, \dots, s$. Using MAGMA, we were able to find suitable codes to construct schemes for $n = 24-35, 41-44, 51, 52, 53, 61, 62, 71$. We note that the following cases are still open: for $k = 2$, $n = 6-13$ and for $k = 3$, $n = 8-23$.

7.6 Cheating-immune threshold schemes

In this section, we present and compare two constructions of (t, n) -threshold schemes that are cheating-immune. The first method uses cumulative arrays while the second

method uses generalized cumulative arrays.

7.6.1 Cheating-immune threshold schemes from cumulative arrays

We adopt a technique presented in [61] to construct a perfect secret sharing scheme for any given access structure. This technique combines a threshold scheme with a cumulative array. We are going to replace the threshold scheme by a cheating-immune scheme.

Let Γ be the (t, n) -threshold access structure. The collection $\mathcal{T}$ of all maximal unauthorized sets with respect to Γ contains $l = \binom{n}{t-1}$ elements. Consider the cumulative array $(\tau, \mathcal{T})$ where the mapping $\tau : \mathcal{P} \rightarrow 2^{\mathcal{T}}$ is given by

$$\tau(P_i) = \{T_j : P_i \notin T_j, 1 \leq j \leq l\}.$$

Suppose Σ is an (l, l) -secret sharing scheme over $\mathbb{F}_q$ that is strictly k -cheating-immune. Let f be the defining function of Σ . For simplicity, we first consider the case when there is only one cheater in the reconstruction stage.

To share a secret $K \in \mathbb{F}_q$:

1. The dealer uses the scheme Σ to generate l shares $(y_1, y_2, \dots, y_l)$ for the secret K . This means that $K = f(y_1, y_2, \dots, y_l)$.
2. The dealer then distributes the shares $(y_1, y_2, \dots, y_l)$ among the participants according to the cumulative array $(\tau, \mathcal{T})$ for Γ .

We denote the resulting scheme by Π . Each participant P_i will receive

$$|\tau(P_i)| = \binom{n-1}{t-1}$$

values from the dealer.

Theorem 7.6.1. *Let $k = \binom{n-1}{t-1}$. The scheme Π is a (t, n) -threshold secret sharing scheme over $\mathbb{F}_q$ that is 1-cheating-immune and we have $2t > n + 2$. Furthermore, over $\mathbb{F}_2$, we have*

$$\binom{n}{t-1} \geq 2 \left[\binom{n-1}{t-1} + 1 \right].$$

Proof. It is not difficult to see that Π is a (t, n) -threshold secret sharing scheme. Suppose a subset of participants $B \subseteq \mathcal{P}$ with $|B| \geq t$ submit their shares to the combiner and one of them is a cheater. It follows from the construction that the cheater has k shares of the scheme Σ . Now the cheater may corrupt all or some of those k shares. Since Σ is strictly k -cheating-immune then the probability of successful cheating in the scheme Π is $1/q$. Therefore, Π is 1-cheating-immune.

Since Σ is strictly k -cheating-immune then from Theorem 7.2.3, we have

$$\begin{aligned} \frac{l}{k} > 2 &\iff \frac{\binom{n}{t-1}}{\binom{n-1}{t-1}} > 2 \\ &\iff \frac{n}{n-t+1} > 2 \\ &\iff 2t > n + 2. \end{aligned}$$

The last statement follows from Theorem 7.2.4. □

We note that the construction can be extended to the case of c cheaters where $1 < c < t$. Suppose $P_{i_1}, P_{i_2}, \dots, P_{i_c}$ are cheaters. Then the combiner will use at most

$$k = \left| \bigcup_{j=1}^c \tau(P_{i_j}) \right|$$

corrupted shares in the computation of the secret. We have

$$\left| \bigcup_{j=1}^c \tau(P_{i_j}) \right| = \sum_{j=1}^c (-1)^{j+1} \binom{c}{j} \binom{n-j}{t-1}.$$

Theorem 7.6.2. *Let*

$$k = \sum_{j=1}^c (-1)^{j+1} \binom{c}{j} \binom{n-j}{t-1}.$$

The scheme Π is a (t, n) -threshold secret sharing scheme over $\mathbb{F}_q$ that is strictly c -cheating-immune and we have

$$\binom{n}{t-1} \geq 2 |\cup_{j=1}^c \tau(P_{i_j})|.$$

Furthermore, over $\mathbb{F}_2$, we have

$$\binom{n}{t-1} \geq 2 [|\cup_{j=1}^c \tau(P_{i_j})| + 1].$$

The main drawback of the construction is that the size of the shares of the scheme Π is generally large. In addition, the construction is only applicable to some values of t and n .

7.6.2 Cheating-immune threshold schemes from generalized cumulative arrays

We now consider using generalized cumulative arrays to construct threshold schemes that are cheating-immune. Generalized cumulative arrays were proposed in [82] in relation to block cipher sharing. The following definition is also from [74].

Definition 7.6.3. *Suppose Γ is an access structure on $\mathcal{P}$. A generalized cumulative array (GCA) $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ for Γ is a collection of finite sets $X_1, \dots, X_l$, where each X_i is associated with a mapping $\tau_i : \mathcal{P} \rightarrow 2^{X_i}$, such that for $I \subseteq \mathcal{P}$ we have*

$$\bigcup_{P \in I} \tau_i(P) = X_i \text{ for some } i \ (1 \leq i \leq l), \text{ if and only if } I \in \Gamma.$$

As shown in [74], we can build a secret sharing scheme realizing a given access structure Γ using a GCA as follows. Let $\mathcal{K}$ be an abelian group and suppose $X_i = \{x_1^{(i)}, \dots, x_{d_i}^{(i)}\}$, $1 \leq i \leq l$. Suppose the dealer D wants to share a secret $K \in \mathcal{K}$ according to the access structure Γ . The dealer D first constructs l (d_i, d_i) -threshold schemes for the secret K . For $1 \leq i \leq l$, suppose the shares generated for the i th scheme are $y_1^{(i)}, \dots, y_{d_i}^{(i)}$. The dealer then distributes the shares according to $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$, that is, the final share of P is

$$s_i = \bigcup_{j=1}^l \left\{ y_j^{(i)} : x_j^{(i)} \in \tau_i(P) \right\}.$$

It follows from the construction that the total number of shares generated by the dealer is equal to $\sum_{i=1}^l |X_i|$ and the number of shares for each $P \in \mathcal{P}$ is given by $\sum_{i=1}^l |\tau_i(P)|$. Ideally, these two values should be minimized. We say that a GCA $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ for Γ is *minimal* if for any GCA $(\tau'_1, \dots, \tau'_{l'}, X'_1, \dots, X'_{l'})$ for Γ , we have $\sum_{i=1}^l |X_i| \leq \sum_{i=1}^{l'} |X'_i|$. Unlike in the case of cumulative arrays, it is not yet known how to construct a minimal GCA for a given access structure.

In this work, we will use a method of obtaining GCAs using perfect hash families [82]. First, we recall the concept of a perfect hash family.

Definition 7.6.4. [74] Let $A = \{1, \dots, n\}$, $B = \{1, \dots, m\}$ and $\mathcal{H}$ be a set of functions from A to B . We say that $\mathcal{H}$ is an (n, m, t) -perfect hash family if for any subset $I \subseteq A$ with $|I| = t$, there exists at least one $h \in \mathcal{H}$ such that h restricted to I is one-to-one. When $|\mathcal{H}| = l$, we refer to $(A, B, \mathcal{H})$ as a $PHF(l; n, m, t)$. When $m = t$, we call it a *minimal perfect hash family*.

Suppose $(A, B, \mathcal{H})$ is a $PHF(l; n, m, t)$ with $\mathcal{H} = \{h_1, \dots, h_l\}$. For each h_i , we associate a t -set $X_i = \{x_1^{(i)}, \dots, x_t^{(i)}\}$. Now, define a function τ_i from $\mathcal{P}$ to 2^{X_i} such

that $\tau_i(P_j) = \left\{ x_{h_i(j)}^{(i)} \right\}$. For any t -subset of $\mathcal{P}$, we can see that there exists an i such that h_i restricted to it is one-to-one. Any subset of $\mathcal{P}$ with less than t participants can only obtain at most $t-1$ elements in X_i for all $1 \leq i \leq l$. Hence, $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ is a GCA for the (t, n) -threshold access structure.

Example 7.6.5. An algorithm to construct a $PHF(l; n, t, t)$ where $l = \binom{n - \lceil \frac{t}{2} \rceil}{\lceil \frac{t}{2} \rceil}$ can be found in [7]. If we set $t = 4, n = 5$ then we obtain a $PHF(3; 5, 4, 4)$ with $\mathcal{H} = \{h_1, h_2, h_3\}$ (shown in Table 7.6.1).

Table 7.6.1: A $PHF(3; 5, 4, 4)$

	1	2	3	4	5
h_1	1	2	3	4	4
h_2	1	2	2	3	4
h_3	2	1	2	3	4

We then get a GCA $(\tau_1, \tau_2, \tau_3, X_1, X_2, X_3)$ for the $(4, 5)$ -threshold access structure with $l = 3$ and $|X_i| = 4$ where

$$X_1 = \left\{ x_1^{(1)}, \dots, x_4^{(1)} \right\}, X_2 = \left\{ x_1^{(2)}, \dots, x_4^{(2)} \right\}, X_3 = \left\{ x_1^{(3)}, \dots, x_4^{(3)} \right\}$$

and the mappings are given in Table 7.6.2.

We are now ready to construct cheating-immune threshold scheme using GCA. Suppose $(A, B, \mathcal{H})$ is a $PHF(l; n, m, t)$ with $\mathcal{H} = \{h_1, \dots, h_l\}$ and consider the GCA $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ (obtained from the PHF) for the (t, n) -threshold access structure Γ . Let Σ be a (t, t) -secret sharing scheme over $\mathbb{F}_q$ that is strictly k -cheating-immune. Let f be the defining function of Σ .

Table 7.6.2: The mappings of a GCA for the $(4, 5)$ -threshold access structure

	P_1	P_2	P_3	P_4	P_5
τ_1	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_4^{(1)}$	$x_4^{(1)}$
τ_2	$x_1^{(2)}$	$x_2^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$x_4^{(2)}$
τ_3	$x_2^{(3)}$	$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$x_4^{(3)}$

To share a secret $K \in \mathbb{F}_q$:

1. The dealer generates l sets of shares using the scheme Σ . For $1 \leq i \leq l$, we denote the i th set of shares by $Y_i = (y_1^{(i)}, \dots, y_t^{(i)})$.
2. The dealer then distributes the shares among the participants according to the generalized cumulative array $(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ for Γ .

We denote the resulting scheme by Π . Each participant P will receive only one share from each set of shares of the scheme Σ . Thus, the number of shares of each participant P is equal to l .

Theorem 7.6.6. *The scheme Π is a (t, n) -threshold secret sharing scheme over $\mathbb{F}_q$ that is strictly k -cheating-immune and we have $2k < t$. Furthermore, over $\mathbb{F}_2$, we have $2k < t - 1$.*

Proof. Suppose a subset of participants $B \subseteq \mathcal{P}$ with $|B| \geq t$ submit their shares to the combiner. If all of the shares submitted are valid then there will be at least one valid set Y_i of t shares of the scheme Σ . The combiner can use Y_i to compute the secret.

Suppose there are k cheaters in B . It follows from the construction that for each set of shares Y_i , $1 \leq i \leq l$, there can only be at most k corrupted shares. Since Σ is

strictly k -cheating-immune then the probability of successful cheating in the scheme Π is $1/q$. Therefore, Π is also strictly k -cheating-immune.

The bounds on the number of cheaters follow from Theorems 7.2.3 and 7.2.4. $\square$

To see the advantage (over cumulative arrays) of using GCAs, first we define

$$M_{(t,n)} = \min \left\{ \sum_{i=1}^l |X_i| \right\}$$

where the minimum is taken over all $\text{GCA}(\tau_1, \dots, \tau_l, X_1, \dots, X_l)$ for the (t, n) -access structure. In [74], the following result is proved.

Theorem 7.6.7.

$$\frac{c(t-1)^2 \log n}{\log(t-1)} \leq M_{(t,n)} \leq \lceil t^2 e^t \log n \rceil,$$

where c is a constant that is approximately $1/2$.

The theorem implies that for a fixed t that is small compared to n , we have $\sum_{i=1}^l |X_i|$ is $O(\log n)$. Recall that in the case of cumulative arrays ($l = 1$), we have $\sum_{i=1}^l |X_i| = |X_1| \geq \binom{n}{t-1}$. Since $\sum_{i=1}^l |X_i|$ is an upper bound on $\sum_{i=1}^l |\tau_i(P)|$ (the number of shares given to participant P), then by using GCA it is also possible to reduce the size of the shares to $O(\log n)$ in the scheme Π .

Another advantage of using GCA over CA is that the construction requires a strictly cheating-immune scheme Σ with smaller parameters. For example, suppose we want to obtain a (t, n) -threshold scheme that is 1-cheating-immune. The construction using CA requires that Σ is an (l, l) -threshold scheme that is strictly k -cheating-immune where $l = \binom{n}{t-1}$ and $k = \binom{n-1}{t-1}$. The construction using GCA only requires Σ is a (t, t) -threshold scheme that is 1-cheating-immune.

Example 7.6.8. *Suppose we want to construct a $(4, 5)$ -threshold scheme Π over $\mathbb{F}_2$ that is 1-cheating-immune. We need the following:*

1. *a GCA for the $(4, 5)$ -threshold access structure*
2. *a $(4, 4)$ -secret sharing scheme Σ that is 1-cheating-immune*

A GCA $(\tau_1, \tau_2, \tau_3, X_1, X_2, X_3)$ for the $(4, 5)$ -threshold access structure is presented in Example 7.6.5 while a 1-CI $(4, 4)$ -SSS (together with a table of distribution of shares) is given in Example 7.2.5.

Suppose the secret is $K = 0$. Using Table 7.2.1, we can define $X_1 = \{0, 1, 0, 0\}$, $X_2 = \{0, 1, 1, 0\}$ and $X_3 = \{0, 1, 1, 1\}$. We then distribute the shares according to the GCA for the $(4, 5)$ -threshold access structure. Thus, P_1 receives $\{0, 0, 1\}$, P_2 receives $\{1, 1, 0\}$, P_3 receives $\{0, 1, 1\}$, P_4 receives $\{0, 1, 1\}$, and P_5 receives $\{0, 0, 1\}$.

If we want to use cumulative arrays then the construction requires a strictly 4-cheating-immune $(10, 10)$ -scheme. Hence, each participant will receive 4 values.

7.7 Conclusion

In this chapter, we have presented new constructions of cheating-immune secret sharing schemes using binary linear codes and cumulative arrays. We modified two known constructions of certain Boolean functions using linear codes so that they can be used to obtain new cheating-immune and strictly cheating-immune (n, n) -secret sharing schemes. We were able to obtain new schemes by searching for suitable codes.

We also constructed cheating-immune (t, n) -threshold schemes using two approaches. The first approach uses cumulative arrays while the second one uses generalized cu-

mulative arrays from perfect hash families. The approach using GCAs can produce schemes with better parameters than the schemes obtained using the cumulative arrays. The resulting schemes are not really good since in general, the size of the shares are very large (exponential in the number of participants).

We already mentioned that there are still many open cases in the construction of cheating-immune and strictly cheating-immune schemes for the (n, n) -threshold access structure. For the (t, n) -threshold access structure, the main question is the existence of cheating-immune schemes with smaller shares than the ones we obtained.

Chapter 8

Concluding Remarks

In this chapter, we summarize the results in this thesis and discuss future research directions by giving open problems in addition to the ones already mentioned at the end of each chapter.

We studied two classes of secret sharing schemes constructed using linear codes in this thesis. The first class consists of the secret sharing schemes obtained using the approach proposed by Karnin, Greene and Hellman, by Bertilsson and Ingemarsson, and by Massey. We dealt with three topics involving this class of secret sharing schemes.

The first topic concerns secret sharing schemes based on the following codes from complete graphs: cut-set code, cycle code, dual of extended cycle code and dual of augmented extended cycle code. Our contributions on this topic are as follows:

- We described the access structures of the schemes based on the cut-set and cycle codes. We also showed that the access structures are not Q_2 , implying that the schemes cannot be multiplicative.

- For the access structure based on odd cycles (realized by the scheme based on the dual of the extended cycle code), we proved that it does not admit any ideal multiplicative LSSS.
- We also demonstrated that the ideal LSSS based on the dual of an augmented extended cycle code is multiplicative in some cases.

On the subject of multiplicative LSSS, the main open problem is the efficient construction of strongly multiplicative LSSS. Another interesting problem is the characterization of access structures which admit ideal multiplicative LSSS.

The second topic is about the access structure of a generalization of secret sharing schemes based on linear codes. In this generalization, the secret is extended from one coordinate to several coordinates. The resulting scheme is non-perfect but has a higher information rate than perfect schemes. We list down our results on this topic.

- We presented a characterization of the groups in the access structure using the dual code. This is an extension of a well-known result by Massey.
- We also showed how the g -fold joint weight enumerator can be used to describe the access structure of schemes based on binary codes. The g -fold joint weight enumerator gives information on the number of authorized groups of certain sizes. As a particular case, we described the access structure of schemes based on binary self-dual codes with symmetry properties.

We note that we only considered binary codes. Thus, it would be interesting to study the case of linear codes over non-binary fields.

The third topic is on the maximum number of minimal codewords in binary codes. This quantity was studied before in the context of cycles in graphs and of circuits in matroids. We achieved the following results:

- Given the length n and dimension k , we proved some upper and lower bounds on this maximum number.
- We computed explicit bounds and some exact values for $n \leq 15$ and $1 \leq k \leq 15$. For the same range of n and k , we determined the exact values for cycle codes of graphs.
- We also analyzed the asymptotic behavior of this maximum number of minimal codewords. We were able to determine the asymptotic value for cycle codes.

The determination of the minimal codewords of linear codes is the main open problem on the topic of minimal codewords. It would be worthwhile also to consider the case of codes over other fields and the minimum number of minimal codewords.

The second class of secret sharing schemes constructed using linear codes that we studied consists of cheating-immune schemes. The following are the results that we obtained:

- We constructed new cheating-immune and strictly cheating-immune (n, n) -secret sharing schemes. The methods that we used are modifications of two known constructions of Boolean functions satisfying some cryptographic criteria. The methods use binary linear codes.
- We constructed cheating-immune (t, n) -threshold schemes using two approaches. The first approach uses cumulative arrays while the second one uses generalized

cumulative arrays from perfect hash families. These schemes are new but the size of the shares tends to be very large.

The main open problem on cheating-immune secret sharing schemes is the construction of such schemes for general access structures.

Bibliography

- [1] E. Agrell. Voronoi regions for binary linear block codes. *IEEE Trans. Inf. Theory*, 42:310–316, 1996.
- [2] E. Agrell. On the voronoi neighbor ratio for binary linear codes. *IEEE Trans. Inf. Theory*, 44(7):3064–3072, 1998.
- [3] A. Alamadhi, R. Aldred, R. dela Cruz, P. Solé, and C. Thomassen. The maximum number of minimal codewords in an $[n, k]$ -code. Submitted to *Discrete Mathematics*.
- [4] A. Alamadhi, R. Aldred, R. dela Cruz, P. Solé, and C. Thomassen. The maximum number of minimal codewords in long codes. *Discrete Applied Mathematics*, 161(3):424–429, 2013.
- [5] A. Ashikhmin and A. Barg. Minimal vectors in linear codes. *IEEE Trans. Inf. Theory*, 44(5):2010–2017, 1998.
- [6] A. Ashikhmin, A. Barg, G. Cohen, and L. Huguët. Variations on minimal codewords in linear codes. In *Proc. of the 11th Int. Symp. on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-11*, pages 96–105. Springer-Verlag, 1995.

- [7] M. Atici, S. Magliveras, D. Stinson, and W. Wei. Some recursive constructions for perfect hash families. *Journal of Combinatorial Designs*, 4:353–363, 1996.
- [8] S. Barwick, W.-A. Jackson, and K. Martin. Updating the parameters of a threshold scheme by minimal broadcast. *IEEE Trans. Inf. Theory*, 51(2):620–633, 2005.
- [9] A. Beimel. *Secure schemes for secret sharing and key distribution*. PhD thesis, Technion-Israel Inst. Technol., Haifa, Israel, 1996.
- [10] A. Beimel. Secret Sharing Schemes: A Survey. In *Proc. IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46, 2011.
- [11] A. Beimel and B. Chor. Universally ideal secret-secret sharing schemes. *IEEE Trans. Inf. Theory*, 40(3):786–794, 1994.
- [12] A. Beimel and B. Chor. Secret sharing with public reconstruction. *IEEE Trans. Inf. Theory*, 44(5):1887–1896, 1998.
- [13] M. Bellare and P. Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proc. of the 14th ACM conference on Computer and Communications Security, CCS ‘07*, pages 172–184, New York, NY, USA, 2007. ACM.
- [14] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. of the 20th Annual ACM Symposium on Theory of Computing, STOC ‘88*, pages 1–10, New York, NY, USA, 1988. ACM.

- [15] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory*, 24:384–386, 1978.
- [16] M. Bertilsson and I. Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 67–79. Springer Berlin Heidelberg, 1993.
- [17] A. Betten. Distance optimal indecomposable codes over $GF(2)$. Available online at http://www.math.colostate.edu/~betten/research/codes/GF2/codes_GF2.html.
- [18] B. Blakley, G. R. Blakley, A. H. Chan, and J. L. Massey. Threshold schemes with disenrollment. In *Advances in Cryptology - CRYPTO 92*, volume 740 of *Lecture Notes in Computer Science*, pages 540–548. Springer Berlin Heidelberg, 1993.
- [19] G. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 Natl. Computer Conf.*, volume 48, pages 313–317, New York, USA, 1979.
- [20] G. Blakley and G. Kabatianskii. Linear algebra approach to secret sharing schemes. In *Error Control, Cryptology, and Speech Compression*, volume 829 of *Lecture Notes in Computer Science*, pages 33–40. 1994.
- [21] G. R. Blakley and C. Meadows. Security of ramp schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 242–268. Springer Berlin Heidelberg, 1985.

- [22] C. Blundo, A. Cresti, A. D. Santis, and U. Vaccaro. Fully dynamic secret sharing schemes. In *Advances in Cryptology - CRYPTO 93*, volume 773 of *Lecture Notes in Computer Science*, pages 110–125. Springer Berlin Heidelberg, 1994.
- [23] C. Blundo, A. D. Santis, and U. Vaccaro. Efficient sharing of many secrets. In *STACS 93*, volume 665 of *Lecture Notes in Computer Science*, pages 692–703. Springer Berlin Heidelberg, 1993.
- [24] Y. Borissov and N. Manev. Minimal codewords in linear codes. *Serdica Mathematical Journal*, 30(2-3):303–324, 2004.
- [25] W. Bosma, J. Cannon, and C. Playoust. The Magma Algebra System I: The User Language. *Journal of Symbolic Computation*, 24:235–265, 1997.
- [26] A. Braeken, V. Nikov, and S. Nikova. On cheating immune secret sharing. In *Proc. 25th Symposium on Information Theory in the Benelux*, pages 113–120, 2004.
- [27] E. Brickell. Some ideal secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 6:105–113, 1989.
- [28] E. Brickell and D. Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4:123–134, 1991.
- [29] P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation-immune functions. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 86–100, 1992.

- [30] R. Capocelli, A. D. Santis, L. Gargano, and U. Vaccaro. A note on secret sharing schemes. In *Sequences II*, pages 335–344. Springer New York, 1993.
- [31] R. M. Capocelli, A. D. Santis, L. Gargano, and U. Vaccaro. On the size of shares for secret sharing schemes. *Journal of Cryptology*, 6(3):157–167, 1993.
- [32] H. Chen and R. Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In *Advances in Cryptology - CRYPTO '06*, volume 4117 of *Lecture Notes in Computer Science*, pages 521–536. Springer-Verlag, 2006.
- [33] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, and V. Vaikuntanathan. Secure Computation from Random Error Correcting Codes. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 329–346. Springer-Verlag, 2007.
- [34] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proc. of the 26th Annual Symposium on Foundations of Computer Science, SFCS '85*, pages 383–395, Washington, DC, USA, 1985. IEEE Computer Society.
- [35] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, second edition, 2006.
- [36] R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret-sharing schemes. In *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer-Verlag, 2000.

- [37] R. Cramer, V. Daza, I. Gracia, J. Urroz, G. Leander, J. Martí-Farré, and C. Padró. On codes, matroids, and secure multi-party computation from linear secret-sharing schemes. *IEEE Trans. Inf. Theory*, 54(6):2644–2657, 2008.
- [38] C. C. D. Chaum and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. of the 20th Annual ACM Symposium on Theory of Computing, STOC '88*, pages 11–19, New York, NY, USA, 1988. ACM.
- [39] P. D’Arco, W. Kishimoto, and D. Stinson. Properties and Constraints of Cheating-Immune Secret Sharing Schemes. *Discrete Applied Mathematics*, 154:219–233, 2006.
- [40] R. dela Cruz, A. Meyer, and P. Solé. An extension of massey scheme for secret sharing. In *Proceedings of IEEE Information Theory Workshop 2010*, pages 1–5, Dublin, Ireland, 2010.
- [41] R. dela Cruz and H. Wang. Cheating-immune secret sharing schemes from codes and cumulative arrays. *Cryptography and Communications*. To appear.
- [42] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer New York, 1990.
- [43] Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In *Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 277–287. Springer Berlin Heidelberg, 2005.

- [44] C. Ding and J. Yuan. Covering and Secret Sharing with Linear Codes. In *Discrete Mathematics and Theoretical Computer Science*, volume 2731 of *Lecture Notes in Computer Science*, pages 11–25. Springer-Verlag, 2003.
- [45] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, 1993.
- [46] G. Dósa, I. Szalkai, and C. Laflamme. The maximum and minimum number of circuits and bases of matroids. *Pure Mathematics and Applications*, 15(4):383–392, 2004.
- [47] S. Dougherty, M. Harada, and M. Oura. Note on g -fold joint weight enumerators of self-dual codes over $\mathbb{Z}_k$. *Appl. Algebra Eng. Commun. Comput.*, 11:437–445, 2001.
- [48] S. Dougherty, S. Mesnager, and P. Solé. Secret-sharing schemes based on self-dual codes. In *Proceedings of IEEE Information Theory Workshop, ITW 2008*, Porto, Portugal, 2008.
- [49] R. Entringer and P. Slater. On the maximum number of cycles in a graph. *Ars Combinatoria*, 11:289–294, 1981.
- [50] S. Fehr. Efficient construction of the dual span program. Available online at <http://homepages.cwi.nl/~fehr/publications.html>, May 1999.
- [51] A. Gál. *Combinatorial Methods in Boolean Function Complexity*. PhD thesis, University of Chicago, 1995.

- [52] Y. Gao and R. dela Cruz. Secret sharing schemes based on graphical codes. Submitted to *Advances in Mathematics of Communications*.
- [53] A. Gerards and A. Schrijver. Signed graph - regular matroids - grafts. Research memorandum, Faculteit der Economische Wetenschappen, Tilburg University, 1986.
- [54] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proc. 19th Annual ACM Symposium on Theory of Computing, STOC '87*, pages 218–229, New York, USA, 1987.
- [55] X. Guo-Zhen and J. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Trans. Inf. Theory*, IT-34(3):569–571, 1988.
- [56] S. Hakimi and J. Bredeson. Graph-theoretic error-correcting codes. *IEEE Trans. Inf. Theory*, IT-14:584–591, 1968.
- [57] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In *Advances in Cryptology - CRYPTO 95*, volume 963 of *Lecture Notes in Computer Science*, pages 339–352. Springer Berlin Heidelberg, 1995.
- [58] W. Huffman. The biweight enumerator of self-orthogonal binary codes. *Discrete Mathematics*, 26:129–143, 1979.
- [59] W. C. Huffman and V. Pless. *Fundamentals of Error-Correcting Codes*. Cambridge University Press, August 2003.

- [60] T.-Y. Hwang. Decoding linear block codes for minimizing word error rate. *IEEE Trans. Inf. Theory*, 25(6):733–737, 1979.
- [61] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *IEEE Globecom 1987*, pages 99–102, 1987.
- [62] W.-A. Jackson and K. Martin. Cumulative Arrays and Geometric Secret Sharing Schemes. In *Advances in Cryptology - Auscrypt '92*, volume 718 of *Lecture Notes in Computer Science*, pages 48–55, 1993.
- [63] W.-A. Jackson and K. Martin. A combinatorial interpretation of ramp schemes. *Australasian Journal of Combinatorics*, 14:51–60, 1996.
- [64] W.-A. Jackson, K. Martin, and C. O’keefe. Mutually Trusted Authority-Free Secret Sharing Schemes. *Journal of Cryptology*, 10:261–289, 1997.
- [65] D. Jungnickel and S. Vanstone. Graphical Codes Revisited. *IEEE Trans. Inf. Theory*, 43(1):136–146, 1997.
- [66] M. Karchmer and A. Wigderson. On span programs. In *Proc. 8th IEEE Structure in Complexity Theory*, pages 102–111, 1993.
- [67] E. Karnin, J. Greene, and M. Hellman. On secret sharing systems. *IEEE Trans. Inf. Theory*, 29(1):35–41, 1983.
- [68] E. Kasper, S. Nikova, and V. Nikov. Strongly multiplicative hierarchical threshold secret sharing. In *Proc. 2nd Int. Conf. on Information Theoretic Security*, volume 4883 of *Lecture Notes in Computer Science*, pages 148–168. Springer-Verlag, 2007.

- [69] K. Kurosawa, S. Obana, and W. Ogata. t -cheater identifiable (k, n) threshold secret sharing schemes. In *Advances in Cryptology - CRYPTO 95*, volume 963 of *Lecture Notes in Computer Science*, pages 410–423. Springer Berlin Heidelberg, 1995.
- [70] K. Kurosawa and K. Okada. Combinatorial lower bounds for secret sharing schemes. *Information Processing Letters*, 60(6):301–304, 1996.
- [71] K. Kurosawa and T. Satoh. Design of SAC/PC(l) of order k Boolean functions and three other cryptographic criteria. In *Advances in Cryptology - EURO-CRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 434–449, 1997.
- [72] C. Li, S. Ling, and L. Qu. On the covering structures of two classes of linear codes from perfect nonlinear functions. *IEEE Trans. Inf. Theory*, 55:70–82, 2009.
- [73] M. Liu, L. Xiao, and Z. Zhang. Multiplicative linear secret sharing schemes based on connectivity of graphs. *IEEE Trans. Inf. Theory*, 53(11):3973–3978, 2007.
- [74] S. Long, J. Pieprzyk, H. Wang, and D. Wong. Generalised cumulative arrays in secret sharing. *Designs, Codes and Cryptography*, 40:191–209, 2006.
- [75] W. Ma and M. Lee. New methods to construct cheating immune functions. In *Information Security and Cryptology - ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 79–86. Springer Berlin Heidelberg, 2004.

- [76] W. Ma and F. Zhang. New methods to construct cheating immune multisecret sharing scheme. In *Information Security and Cryptology*, volume 3822 of *Lecture Notes in Computer Science*, pages 384–394. Springer Berlin Heidelberg, 2005.
- [77] F. MacWilliams, C. Mallows, and N. Sloane. Generalizations of Gleason’s Theorem on Weight Enumerators of Self-Dual Codes. *IEEE Trans. Inf. Theory*, 18:794–805, 1972.
- [78] K. Martin. *Discrete Structures in the Theory of Secret Sharing*. PhD thesis, University of London, 1991.
- [79] K. Martin. *Untrustworthy Participants in Secret Sharing Schemes*, pages 255–264. 1993.
- [80] K. Martin. Challenging the adversary model in secret sharing schemes. In *Coding and Cryptography II, Proc. of the Royal Flemish Academy of Belgium for Science and the Arts*, pages 45–63, 2008.
- [81] K. Martin, J. Pieprzyk, R. Safavi-Naini, and H. Wang. Changing Thresholds in the Absence of Secure Channels. In *Information Security and Privacy*, volume 1587 of *Lecture Notes in Computer Science*, pages 177–191. Springer Berlin Heidelberg, 1999.
- [82] K. Martin, R. Safav-Naini, H. Wang, and P. Wild. Distributing the encryption and decryption of a block cipher. *Designs, Codes and Cryptography*, 36:263–287, 2005.

- [83] J. L. Massey. Minimal codewords and secret sharing. In *Proc. 6th Joint Swedish-Russian Workshop Inf. Theory*, pages 276–279, Molle, Sweden, 1993.
- [84] J. L. Massey. Some applications of coding theory in cryptography. In P. Farrell, editor, *Codes and Ciphers, Cryptography and Coding IV*, pages 33–47, Esses, England, 1995. Formara Ltd.
- [85] R. McEliece and D. Sarwate. On sharing secrets and reed-solomon codes. *Communications of the ACM*, 24:583–584, 1981.
- [86] B. McKay. nauty - program for computing automorphism groups of graphs and digraphs. Available online at <http://cs.anu.edu.au/~bdm/nauty/>.
- [87] S. Nikova and V. Nikov. On multiplicative secret sharing schemes realizing graph access structures. In *International Workshop on Optimal Codes and Related Topics*, pages 194–199, Balchik, Bulgaria, 2007.
- [88] W. Ogata and K. Kurosawa. Some Basic Properties of General Nonperfect Secret Sharing Schemes. *Journal of Universal Computer Science*, 4(8):690–704, 1998.
- [89] W. Ogata, K. Kurosawa, and D. Stinson. Optimum secret sharing scheme secure against cheating. *SIAM Journal on Discrete Mathematics*, 20(1):79–95, 2006.
- [90] J. Oxley. *Matroid Theory*. Oxford Science Publications, The Clarendon Press, Oxford University Press, New York, 1992.
- [91] C. Padró and I. Gracia. Representing small identically self-dual matroids by self-dual codes. *SIAM J. Discrete Math*, 20:1046–1055, 2006.

- [92] J. Pieprzyk and X. M. Zhang. Cheating Prevention in Secret Sharing over $gf(p^t)$. In *Progress in Cryptology - INDOCRYPT 2001*, volume 2247 of *Lecture Notes in Computer Science*, pages 79–90, 2001.
- [93] J. Pieprzyk and X. M. Zhang. Constructions of Cheating-Immune Secret Sharing. In *Information Security and Cryptology - ICISC 2001*, volume 2288 of *Lecture Notes in Computer Science*, pages 226–243, 2002.
- [94] J. Pieprzyk and X. M. Zhang. On Cheating Immune Secret Sharing. *Discrete Mathematics and Theoretical Computer Science*, 6:253–264, 2004.
- [95] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. 21st Annual ACM Symposium on Theory of Computing, STOC '89*, pages 73–85, New York, USA, 1989.
- [96] A. Renvall and C. Ding. The access structure of some secret sharing schemes. In *Information Security and Privacy*, volume 1172 of *Lecture Notes in Computer Science*, pages 67–78. Springer Berlin Heidelberg, 1996.
- [97] C. P. S. Cabello and G. Sáez. Secret sharing schemes with detection of cheaters for a general access structure. *Designs, Codes and Cryptography*, 25:175–188, 2002.
- [98] R. Safavi-Naini and H. Wang. Secret sharing schemes with partial broadcast channels. *Designs, Codes and Cryptography*, 41(1):5–22, 2006.
- [99] P. D. Seymour. On secret-sharing matroids. *Journal of Combinatorial Theory, Series B*, 56(1):69–73, 1992.

- [100] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
- [101] B. Shankar, K. Srinathan, and C. P. Rangan. Alternative protocols for generalized oblivious transfer. In *Distributed Computing and Networking*, volume 4904 of *Lecture Notes in Computer Science*, pages 304–309. Springer Berlin Heidelberg, 2008.
- [102] G. J. Simmons, W. Jackson, and K. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1(2):230–236, 1991.
- [103] J. Simonis and A. Ashikhmin. Almost affine codes. *Designs, Codes and Cryptography*, 14:179–197, 1998.
- [104] N. J. A. Sloane. Covering arrays and intersecting codes. *Journal of Combinatorial Designs*, 1(1):5163, 1993.
- [105] R. Stanley. Invariants of finite groups and their applications to combinatorics. *Bull. AMS*, 3:475–497, 1979.
- [106] D. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2:357–390, 1992.
- [107] T. Tassa. Generalized oblivious transfer by secret sharing. *Designs, Codes and Cryptography*, 58:11–21, 2011.
- [108] M. Tompa and H. Woll. How to share a secret with cheaters. *Journal of Cryptology*, 1:133–138, 1988.

- [109] M. van Dijk. A linear construction of secret sharing schemes. *Designs, Codes and Cryptography*, 12:161–201, 1997.
- [110] M. van Dijk. *Secret Key Sharing and Secret Key Generation*. PhD thesis, Technische Universiteit Eindhoven, 1997.
- [111] H. Wang and D. Wong. On secret reconstruction in secret sharing schemes. *IEEE Trans. Inf. Theory*, 54(1):473–480, 2008.
- [112] Y. Wei and Y. Hu. New construction of resilient functions with satisfying multiple cryptographic criteria. In *Proc. of the 3rd Int. Conf. on Information Security InfoSecu '04*, pages 175–180. ACM, 2004.
- [113] D. J. A. Welsh. *Matroid Theory*. Academic Press, London, 1976.
- [114] D. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2001.
- [115] A. Yao. Protocols for secure computation. In *Proc. 23rd IEEE Symp. Foundation of Computer Science, FOCS '82*, pages 160–164, Chicago, USA, 1982.
- [116] K. Yasunaga and T. Fujiwara. Determination of the local weight distribution of binary linear block codes. *IEEE Trans. Inf. Theory*, 52:4444–4454, 2006.

List of Works

Below is the list of papers that are included in this thesis:

1. R. dela Cruz and H. Wang. Cheating-immune secret sharing schemes from codes and cumulative arrays. *Cryptography and Communications*, 5(1):67-83, March 2013.
2. A. Alahmadi, R.E.L. Aldred, R. dela Cruz, P. Solé and C. Thomassen. The maximum number of minimal codewords in an $[n, k]$ -code. To appear in *Discrete Mathematics*.
3. A. Alahmadi, R.E.L. Aldred, R. dela Cruz, P. Solé and C. Thomassen. The maximum number of minimal codewords in long codes. *Discrete Applied Mathematics*, 161(3):424-429, February 2013.
4. Y. Gao and R. dela Cruz. Secret sharing schemes based on graphical codes. Submitted.
5. R. dela Cruz, A. Meyer, and P. Solé. An extension of Massey scheme for secret sharing. In *Proceedings of IEEE Information Theory Workshop 2010*, Dublin, Ireland, 2010.

List of Figures

4.3.1 K_4	77
6.3.1 $\mu(R)$ versus R	119

List of Tables

6.2.1 $g(n)$ for $3 \leq n \leq 15$	110
6.2.2 $W(n, k)$ for $3 \leq n \leq 15, 1 \leq k \leq 9$	111
6.2.3 $W(n, k)$ for $10 \leq n \leq 15, 10 \leq k \leq 15$	112
6.2.4 $W_g(n, k)$ for $1 \leq k \leq 10, 3 \leq n \leq 15$	113
7.2.1 Share distribution of a 1-CI $(4, 4)$ -SSS	128
7.6.1 A $PHF(3; 5, 4, 4)$	143
7.6.2 The mappings of a GCA for the $(4, 5)$ -threshold access structure . . .	144