Chapter 1

Foundations

The notion of secret sharing was independently introduced in 1979 by Shamir [36] and Blakley [7]. A secret-sharing scheme (also called SSS) is a method by which one can distribute shares of a secret to a set of participants in such a way that only specified groups of participants can reconstruct the secret by pooling their shares. Secret-sharing schemes are widely used in cryptographic protocols. In particular, it is common in the field of key management and key distribution that keys are shared in fragments to make an attack more difficult. For example, it may be useful for a node A in a network to obtain a private key. This key can be given to him by a trusted third party who then becomes key escrow. In some networks such as *ad hoc* networks, which do not have a trusted third party, this key may be provided by a set of nodes, each of which provides a fragment of the key. Thus, in the absence of collision of these nodes, only node A will know its key which will be a function of the fragments.

There are many other applications to secret sharing, like secure multiparty computation which is perhaps the application with the most potential. Secure Multiparty Computation (SMPC) is a cryptographic technique that allows multiple parties to jointly and securely compute a function while keeping their inputs private. This can be useful, for example, to create secure, privacy-preserving machine learning. It has been introduced in the early 1980s by Yao [45] as a solution to the famous Millionaires' problem: two millionaires want to determine who is the richest without revealing their personal wealth. With theoretical constructs dating back to the 1980s, there have been

substantial improvements in algorithmic and engineering designs over the past decade to improve performance in terms of computation times [18]. In fact, ARPA mentions that the overall performance of sMPCs has increased by four to five orders of magnitude in the last decade alone, representing drastic improvements. As a result, applications of sMPCs are no longer relegated to theoretical designs and are now firmly rooted in the practical world. Another important cryptographic application, which can be related to sMPC, is the so-called oblivious transfer [23]. In such a protocol, a sender transmits an information, selected among several possible sendings, to a recipient, without the sender being able to know the choice of the recipient, nor the recipient being able to know the information he did not request.

In general, secret-sharing schemes are used in many security protocols, and still have great potential both theoretically and practically. Thus, the theoretical appeal and the profusion of applications explain the interest of researchers in the domain.

We are interested, in this document, in a large family of schemes called linear and which have the advantage of being able to use the tools of linear algebra. There are several approaches to building linear secret-sharing schemes. One can for example quote those based on matroids [44]. Matroids can be used to obtain specific properties of secret-sharing schemes. In this book, we will focus on constructions based on error correcting codes defined over the finite field $\mathbb{F}_q$, q being a prime power.

In this chapter, we set the notations that will be used throughout the book and we give the general definitions related to secret-sharing scheme and coding theory. We consider the two main constructions of secret-sharing schemes based on linear codes. For both constructions, we analyze the properties of the underlying codes that allow us to obtain the secret-sharing schemes with the desired parameters. In particular, we show the importance of the so-called Maximum Distance Separable (MDS) codes. Then, we introduce the most popular schemes which can be constructed using a linear code: Shamir [36], Massey [31, 32] and Blakley [7] secret-sharing schemes. Shamir scheme, the best-known secret-sharing scheme, is often presented in terms of univariate polynomials. But we will see that it can also be constructed using a code-based construction, in this case a Reed–Solomon code.

Finally, we introduce multisecret-sharing schemes that can be seen as generalizations of secret-sharing schemes in the sense that these schemes share m arbitrarily related secrets among a set of n participants. It is thus possible to define many thresholds, depending on the secrets. The study of these schemes has been the subject of numerous studies (see for example [1, 4, 11, 13, 13, 14]).

Specific properties of these schemes are detailed in the following chapters.

1.1. Access Structures

A secret-sharing scheme (SSS) involves a *dealer* who detains a *secret*. This dealer distributes its secret to a set of *participants* (also called users or shareholders) in order that each party holds a *share* (or fragment) of that secret. Some special subsets of participants can reconstruct the secret while the other cannot. The groups that can reconstruct the secret are called *qualified* (or sometimes authorized) and the other groups are called *rejected* (or sometimes forbidden).

Let $\mathcal{P} = \{p_1, \ldots, p_n\}$ be a set of participants. A collection $\mathcal{A} \subseteq 2^{\mathcal{P}}$ is *monotone increasing* if $A \in \mathcal{A}$ and $A \subseteq B$ imply that $B \in \mathcal{A}$. This means that any superset of A is also in $\mathcal{A}$. Similarly, a collection $\mathcal{B} \subseteq 2^{\mathcal{P}}$ is *monotone decreasing* if for each set B in $\mathcal{B}$ also each subset of B is in $\mathcal{B}$. A monotone increasing set $\mathcal{A}$ can be efficiently described by the set $\mathcal{A}^-$ consisting of the minimal elements (sets) in $\mathcal{A}$, i.e. the elements in $\mathcal{A}$ for which no proper subset is also in $\mathcal{A}$. Similarly, the set $\mathcal{B}^+$ consists of the maximal elements (sets) in $\mathcal{B}$, i.e. the elements in $\mathcal{B}$ for which no proper superset is also in $\mathcal{B}$.

Let $\mathcal{A} \subseteq 2^{\mathcal{P}}$ be the set of qualified groups of participants and $\mathcal{B} \subseteq 2^{\mathcal{P}}$ be the set of rejected groups. The tuple $(\mathcal{A}, \mathcal{B})$ is called an *access structure* if $\mathcal{A} \cap \mathcal{B} = \emptyset$. It is called monotone if $\mathcal{A}$ is monotone increasing and $\mathcal{B}$ is monotone decreasing. In this case, $(\mathcal{A}^+, \mathcal{B}^-)$ generates $(\mathcal{A}, \mathcal{B})$. Most of the access structures are monotone. However, there are situations where it is more appropriate to consider non-monotone access structures (see for example [29]). In this book, we will only consider monotone access structures which is a natural assumption. In this case, if the set $\{a, b\}$ is qualified to initiate an action, then any superset is also able to do so and on the other hand, if the group composed of a and b is rejected, then participant a (respectively, b) is rejected.

If $\mathcal{A} = \emptyset$, then the secret will remain secret and no one will receive a share. When $\mathcal{A} \cup \mathcal{B} = 2^{\mathcal{P}}$, the access structure is called *complete* and can simply be denoted by $\mathcal{A}$ and $\mathcal{B} = \mathcal{A}^c$, the complement of $\mathcal{A}$.

The dual access structure $(\mathcal{A}^\perp, \mathcal{B}^\perp)$ for $(\mathcal{A}, \mathcal{B})$ is such that

$$\mathcal{A}^\perp = \{A \ : \ A^c \in \mathcal{B}\} \quad \text{and} \quad \mathcal{B}^\perp = \{A \ : \ A^c \in \mathcal{A}\}.$$

It is easy to see that if $\mathcal{A}$ is a complete access structure defined on $\mathcal{P}$, then the dual access $\mathcal{A}^\perp$ is the collection of sets $A \in \mathcal{P}$ such that $\mathcal{P} \setminus A = A^c \in \mathcal{B}$. Moreover, we have $(\mathcal{A}^\perp)^\perp = \mathcal{A}$ and $(\mathcal{B}^\perp)^\perp = \mathcal{B}$.

1.2. Secret-Sharing Schemes and Examples

After having introduced the notion of access structure, it is time to look at the notion of secret-sharing scheme. To do so, we rely on examples.

Suppose a company may want to require the collaboration of (at least) one deputy director and one manager to be able to obtain the key to spend an amount that exceeds a certain threshold. Suppose the key s belongs to $\mathbb{Z}_q$, q being an integer. Let $d_1, \ldots, d_2$ be the deputy directors and let $m_1, \ldots, m_2$ be the managers. Then we have

$$\begin{aligned}\mathcal{A} = \{&\{d_1, m_1\}, \{d_1, m_2\}, \{d_2, m_1\}, \{d_2, m_2\}, \{d_1, m_1, m_2\}, \\ &\{d_2, m_1, m_2\}, \{d_1, d_2, m_1\}, \{d_1, d_2, m_2\}, \{d_1, d_2, m_1, m_2\}\}\end{aligned}$$

and

$$\mathcal{B} = \{\emptyset, \{d_1\}, \{d_2\}, \{m_1\}, \{m_2\}, \{d_1, d_2\}, \{m_1, m_2\}\}.$$

The pair $(\mathcal{A}, \mathcal{B})$ is an access structure since $\mathcal{A} \cap \mathcal{B} = \emptyset$. This access structure is monotone and complete. Suppose it is given to each deputy director the share a (randomly chosen in $\mathbb{Z}_q$) and to each manager the share $s + a$. Then, we can see that all the elements of $\mathcal{A}$ are able to compute s while no element of $\mathcal{B}$ is able to do so. In fact, since a is randomly chosen in $\mathbb{Z}_q$ and is independent of s, share $a + s$ is uniformly distributed over $\mathbb{Z}_q$. This means that $a + s$ takes on every value with equal probability and therefore $a + s$ gives no information on s. On the other hand, a group containing shares a and $a + s$ is able to compute the key s by subtracting share a to share $a + s$.

This scheme can be viewed as a pair (Share, Reconstruct) of protocols (or phases). The sharing phase consists for the dealer P_0 to share the secret s to the participants (here, the deputy directors and the managers each receives a share) and the reconstruction phase consists for the participants to try to reconstruct s. Note that the participants of any set of $\mathcal{B}$ learn nothing about the secret s and moreover the key s can be computed by any set of participants $A \in \mathcal{A}$. Note also that the size of a share is exactly the size of the secret.

This pair of protocols along with the two aforementioned properties realize what we call a secret-sharing scheme (SSS) based on the access structure $(\mathcal{A}, \mathcal{B})$. More generally, we can state the definition of an SSS.

Definition 1.2.1. A *secret-sharing scheme* based on an access structure $(\mathcal{A}, \mathcal{B})$ is a pair (Share, Reconstruct) of protocols such that the protocol Share consists for the dealer to compute the shares of a secret and to distribute them to the participants and the Reconstruct protocol consists for groups of participants to try to reconstruct the secret from their shares. Moreover, the scheme must have the following properties:

- *Privacy*: the participants of any element of $\mathcal{B}$ learn nothing about the secret s.
- *Correctness*: the key s can be computed by any group of participants $A \in \mathcal{A}$.

Note that when $\mathcal{B}^c = \mathcal{A}$ (i.e. the access structure is complete), the secret-sharing scheme is called *perfect*.

A qualified group is *minimal* if none of its proper subset is qualified. A secret-sharing scheme is to be *t-democratic* if every group of t participants is in the same number of minimal qualified groups, where $t \geq 1$. A participant is called a *dictator* if she/he is a member of every minimal qualified group.

An important efficient parameter in secret-sharing scheme is *the size* of the shares compare to the size of the secret. In any perfect secret-sharing scheme, the size of each share is greater than the size of the secret. On the contrary, in non-perfect schemes the size of each share can be smaller than the size of the secret. This can be

an advantage for some applications. A complete access structure $\mathcal{A}$ providing an *ideal* secret-sharing scheme is a scheme such that any participant has only one share with size equal to the size of the secret.

Let us now construct a more complex SSS based on the previous example and similar to [38]. Suppose that the company may want to require the collaboration of at least two deputy directors or at least three managers to be able to obtain the key. It is also required that a deputy director should act as a manager. Therefore, two managers and one deputy director form a qualified group. There are m managers counted from 1 to m and d deputy directors counted from $m+1$ to $m+d$. The director chooses randomly the key $s \in \mathrm{GF}(q)$, where $\mathrm{GF}(q)$ is the Galois field of order q (q being a prime power). To share s among the deputy directors and the managers, the director first selects non-zero distinct elements $\alpha_1, \ldots, \alpha_m \in \mathrm{GF}(q)$. Then he selects $\alpha_{m+1}, \ldots, \alpha_{m+d} \in \mathrm{GF}(q)$ such that they are distinct and different from all values

$$\alpha_i \alpha_j (\alpha_i + \alpha_j)^{-1}, \quad 1 \leq i < j \leq m.$$

Of course, this can happen only if $q \geq 1 + m + m(m-1)/2 + d$ since otherwise the number of elements in the field is not sufficient. The director stores these $m+d$ elements in a public directory. In the next step, the director chooses randomly $a, b \in \mathrm{GF}(q)$, and transmits the share

$$s_i = s + a\alpha_i + b(\alpha_i)^2, \quad \text{where } 1 \leq i \leq m$$

to the ith manager, and share

$$s_i = s + a\alpha_i, \quad \text{where } m+1 \leq i \leq m+d$$

to the ith deputy director. The elements a and b are kept secret by the director.

There are three kinds of groups that are qualified. The groups containing at least two deputy directors, the groups containing two managers and one deputy directors and the groups containing three managers. Suppose managers i and j together with the deputy director k combine their shares in order to compute the key s. They use

the public directory of the director to construct the matrix

$$G = \begin{pmatrix} 1 & 1 & 1 \\ \alpha_i & \alpha_j & \alpha_k \\ (\alpha_i)^2 & (\alpha_j)^2 & 0 \end{pmatrix}$$

and its inverse G^{-1} which exists since $\det(G) = -(\alpha_k(\alpha_j)^2) + (\alpha_k(\alpha_i)^2) + (\alpha_i(\alpha_j)^2) - (\alpha_j(\alpha_i)^2) = (\alpha_i - \alpha_j)(\alpha_k(\alpha_i + \alpha_j) - \alpha_i\alpha_j) \neq 0$.

Then they compute

$$(s_i, s_j, s_k)G^{-1} = (s, a, b),$$

in order to obtain s.

The combination of three managers uses the matrix

$$G = \begin{pmatrix} 1 & 1 & 1 \\ \alpha_i & \alpha_j & \alpha_k \\ (\alpha_i)^2 & (\alpha_j)^2 & (\alpha_j)^2 \end{pmatrix}$$

which also has a non-zero determinant since G is the transpose of a Vandermonde matrix and the α_i are all distinct.

The combination of two deputy directors uses the matrix

$$G = \begin{pmatrix} 1 & 1 \\ \alpha_i & \alpha_j \end{pmatrix},$$

which again has a non-zero determinant.

Now, if two managers i and j combine their shares, can they obtain s? Their shares are given by the entries of the vector

$$s(1,1) + (a,b) \begin{pmatrix} 1 & 1 \\ \alpha_i & \alpha_j \\ (\alpha_i)^2 & (\alpha_j)^2 \end{pmatrix}.$$

The information about the value of s contained in this vector is equal to the information about the value of s contained in the vector

$$s(1,1) \begin{pmatrix} \alpha_i & \alpha_j \\ (\alpha_i)^2 & (\alpha_j)^2 \end{pmatrix}^{-1} + (a,b).$$

We know that the director has chosen a and b randomly in $\mathrm{GF}(q)$ and that s, a and b are statistically independent. Therefore, two deputy

directors cannot obtain any information about s. It can be proved in a similar way that a deputy director and a manager cannot collude to obtain s.

It is interesting to note that the construction of the shares makes use of a matrix. In fact, the transformation from the vector (s, a, b) to the shares $(s_1, \ldots, s_{m+d})$ is linear. It means that for any two secrets s and s' and respective share s_i and s'_i (where $1 \leq i \leq m + d$), the shares $s_i + s'_i$ and λs_i are valid shares for the secrets $s + s'$ and λs, respectively. This property is very common and leads us to state the following definition.

Definition 1.2.2. A secret-sharing scheme is said to be *linear* if for any two secrets s and s' and respective share vectors $(s_1, \ldots, s_n)$ and $(s'_1, \ldots, s'_n)$, the vectors $(s_1 + s'_1, \ldots, s_n + s'_n)$ and $(\lambda s_1, \ldots, \lambda s_n)$ are valid share vectors for the secrets $s + s'$ and λs, respectively.

The vast majority of secret-sharing schemes are linear. This is particularly the case when these schemes are based on linear error correcting codes as we will see later.

1.3. Alternative Definitions

There are many definitions of a secret-sharing scheme which are each adapted to a specific objective or mathematical vision. In what follows, we give two more definitions. The first one makes use of the notion of distribution scheme and can be found in [3].

Definition 1.3.1. A *distribution scheme* $\Sigma = \langle \Pi, \mu \rangle$ with domain of secrets K is a pair, where μ is a probability distribution on some finite set R called the set of random strings and Π is a mapping from $K \times R$ to a set of n-tuples $K_1 \times K_2 \times \cdots \times K_n$, where K_j is called the domain of shares of p_j. A dealer distributes a secret according to Σ by first sampling a random string $r \in R$ according to μ, computing a vector of shares $\Pi(k, r) = (s_1, \ldots s_n)$, and privately communicating each share s_j to party p_j. For a set $A \subseteq \{p_1, \ldots, p_n\}$, we denote $\Pi(s, r)_A$ the restriction of $\Pi(s, r)$ to its A-entries.

In a practical implementation of an SSS, it is important for security and efficiency reasons, to keep the size of the shares as small as possible. In order to measure the amount of information that must be

given to the participants, we can use the worst-case *information ratio*, which is the ratio between the maximum size of the shares and the size of the secret. The information ratio of a distribution scheme is

$$\frac{\max_{1\leq j\leq n}\log|K_j|}{\log|K|}.$$

We can also use the *average information ratio* which is the ratio between the arithmetic mean of the size of all shares and the size of the secret. It is defined as

$$\frac{\sum_{1\leq j\leq n}\log|K_j|}{n\log|K|}.$$

The *information rate* is defined as the inverse of the information ratio.

Definition 1.3.2. Let K be a finite set of secrets, where $|K| \geq 2$. A distribution scheme $\langle \Pi, \mu \rangle$ with domain of secrets K is a secret-sharing scheme realizing an access structure $(\mathcal{A}, \mathcal{B})$ if the following requirements hold:

- *Correctness*: The secret k can be reconstructed by any qualified set of participants. That is, for any set $A \in \mathcal{A}$ (where $A = \{p_1, \ldots, p_{|A|}\}$), there exists a reconstruction function $\text{Reconstr}_A : K_{i_1} \times \cdots \times K_{i_{|A|}} \to K$ such that for every $k \in K$,

$$\Pr[\text{Reconstr}_A(\Pi(k, r)_A) = k] = 1. \tag{1.3.1}$$

- *Privacy*: Every rejected set cannot learn anything about the secret (in the information theoretic sense) from their shares. Formally, for any set $B \in \mathcal{B}$, for every two secrets $k_1, k_2 \in K$, and for every possible vector of shares $\langle s_j \rangle_{p_j \in B}$:

$$\Pr[\Pi(k_1, r)_B = \langle s_j \rangle_{p_j \in B}] = \Pr[\Pi(k_2, r)_B = \langle s_j \rangle_{p_j \in B}]. \tag{1.3.2}$$

This definition can be qualified as "strong" in the sense that correctness of the reconstruction with probability 1 is required and distributions $\Pi(k_1, r)_B$ and $\Pi(k_2, r)_B$ must be equal. Indeed, there exist more flexible definitions that simply require that the correctness hold with a high probability and that the distributions be very close.

Secret-sharing schemes can also be defined using the *entropy function* [25]. For this, we assume that a probability distribution over the domain of secrets is known. So let us consider such a probability distribution together with a distribution scheme Σ. This induces, for any set $A \subseteq \{p_1, \ldots, p_n\}$, a probability distribution on the vector of shares corresponding to the participants in A.

The random variable taking values according to this probability distribution is denoted S_A, while S is the random variable denoting the secret. Next, we will see that the privacy and correctness requirements can be formalized using the entropy function. For this, we first need to introduce the notions of entropy.

Let X be a random variable. The support of X, denoted $\text{supp}(X)$, is the set of values x such that $\Pr[X = x] > 0$ and the *entropy* of X is defined as

$$H(X) := \sum_{x \in \text{supp}(X)} \Pr[X = x] \log 1/\Pr[X = x].$$

Intuitively, $H(X)$ measures the amount of uncertainty in X. If X is known, its entropy is equal to 0. Here the notation log means a logarithm of arbitrary base. Very often, the information is represented as bit sequences and logarithm base 2 is used.

Example 1.3.3. Let X be a discrete random variable taking 3 values a, b and c such that $\Pr[X = a] = 1/2$, $\Pr[X = b] = 1/4$, $\Pr[X = a] = 1/4$. Then $H(X) = 1/2 \log_2(2) + 2 * (1/4) \log_2(4) = 1.5$. This also means that encoding X takes 1.5 bits and that the uncertainty of X is of 1.5 bits. Note that in this example we use the logarithm base 2 in order to consider the bit as the information unit.

If X is uniformly distributed over $\text{supp}(X)$, then

$$H(X) = \log |\,\text{supp}(X)|$$

and in general we have $0 \leq H(X) \leq \log |\,\text{supp}(X)|$.

Now, if we consider two random variables X and Y, *the joint entropy* $H(X, Y)$ is a measure of the uncertainty associated with the set of the two variables X and Y. It is defined as

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} Pr(X = x, Y = y) \log[Pr(X = x, Y = y)],$$

where $Pr(X = x, Y = y)$ denotes the joint probability of x and y occurring together. This definition can be generalized to more than two variables but this generalization is useless for our purpose.

The *conditional entropy* $H(X|Y)$ measures the amount of information needed to describe the outcome of a random variable X given that the value of another random variable Y is known. It is defined as

$$H(X|Y) = H(X, Y) - H(Y).$$

If the two variables are independent, we have $H(X|Y) = H(X)$.

Another definition of secret-sharing scheme, equivalent to Definition 1.3.2, is now given using conditional entropy.

Definition 1.3.4. A distribution scheme is a secret-sharing scheme realizing an access structure $(\mathcal{A}, \mathcal{B})$ with respect to a given probability distribution on the secrets, denoted by a random variable S, if the following conditions hold.

- *Correctness*: For every qualified set $A \in \mathcal{A}$,

$$H(S|S_A) = 0.$$

- *Privacy*: For every rejected set $B \in \mathcal{B}$,

$$H(S|S_B) = H(S).$$

Note that the security offered by this model is *unconditional*. This means that it is independent of the amount of computing time and resources that are available when attempting to obtain information about the secret by some unauthorized means.

Blundo *et al.* [10] proved that if a scheme realizes an access structure with respect to one distribution on the secrets, then it realizes the access structure with respect to any distribution with the same support.

We recall that an ideal secret-sharing scheme is a scheme in which the size of the share of each party is exactly the size of the secret. We will see that this is the case of Shamir scheme which is in addition an example of *threshold scheme*. A (k, n) threshold scheme is a scheme for which the qualified sets are all the sets whose size is above a certain *threshold* $k \leq n$, where n is the number of participants.

The notion of *ramp scheme* generalizes that of threshold scheme. A (k, t, n) ramp scheme is such that the qualified sets are all the sets whose size is greater than t. Moreover, any subset of participants of size at most $k-1$ should get no information about the secret and the subsets of participants of size between k and $t-1$ might get some information about the secret.

1.4. Security Models

In order to analyze the security of a secret-sharing scheme with an access structure $(\mathcal{A}, \mathcal{B})$, it is necessary to define the conditions under which the scheme is used. We assume that each participant is connected to a broadcast medium and that the system is synchronized, i.e. the participants can access a common global clock. Moreover, there are secure channels between every two participants and between the dealer and every participant. We also assume that each participant has a local source of randomness. This communication model is called secure-channel and was introduced in [5, 16].

The collaborating participants are asked to execute the protocol correctly but this rule may not be followed. An adversary may corrupt some of the participants subject to certain constraints, for example an upper bound on the total number of corrupted participants. One can distinguish between *passive* and *active* corruption, see Fehr and Maurer [21] for recent results. Passive corruption means that the adversary obtains entire information held by the corrupted participants while they are executing the protocol correctly. These participants are also called *honest* but *curious*. Active corruption means that the adversary takes full control of the corrupted participants. In this case, the protocol may not be correctly executed by some corrupted participants. To be more precise, let D be the set of all the corrupted participants and let $A \subseteq D$ be the set of actively corrupted participants. The set $D \setminus A$ is the set of honest but curious. The adversary is characterized by a *privacy structure* $\mathcal{A} \subseteq 2^{\mathcal{P}}$ and an *adversary structure* $\mathcal{A}_A \subseteq \mathcal{A}$. The adversary can corrupt participants passively or actively as long as they belong to the set D. Active corruption is strictly stronger than passive corruption.

When the set of corrupted participants is chosen once and for all before the protocol starts, the adversary is called *static*.

When the set of corrupted participants is not static, the adversary is called *adaptive*. This means that the adversary can at any time during the protocol choose to corrupt a new participant based on the information he has (as long as this participant belongs to the set of corrupted participants). This model is known as the *mixed* adversary model.

Standard secret-sharing schemes, like for example Shamir schemes, that will be presented in the following section, assume that all participants perform the protocol correctly. Therefore, their security does not include active attacks. However, there are schemes that allow participants to verify their shares. More formally, *verifiable secret sharing* ensures that even if the dealer is malicious there is a well-defined secret that the participants can later reconstruct. The concept of verifiable secret sharing (VSS) was first introduced in 1985 by Chor *et al.* in [17]. All the schemes considered in the following are standard schemes and therefore assume that the protocol is executed correctly.

1.5. Shamir Scheme and Applications

Shamir's threshold secret-sharing scheme [36] is probably the most known secret-sharing scheme and perhaps the most used. It has been introduced by Shamir in 1979 and is still used in many security or network protocols, generally to distribute keys.

The qualified sets are all the sets whose size is greater than a certain *threshold* $t \leq n$, where n is the number of participants. The scheme is usually called Shamir's (t, n)-threshold scheme and its access structure is defined as

$$\mathcal{A} = \{A \subseteq \{p_1, \ldots, p_n\} : |A| \geq t\}.$$

The idea of Shamir is smart albeit very simple: it is based on the fact that a univariate polynomial $y = f(x)$ of degree $t-1$ is uniquely defined by t points (x_i, y_i) with distinct x_i. The knowledge of less than t points gives no information on the polynomial. The secret is the constant term of the polynomial and the shares are the points of the polynomials. The reconstruction of the secret uses *Lagrange interpolation*.

Protocol 1 details the scheme for a prime field (generalization for any field is straightforward).

Protocol 1: Shamir's (t, n) threshold Scheme (in $\mathbb{Z}_p$)

Input: a trusted dealer distributes shares of a secret S to n participants.
Goal: any group of t participants which pool their shares can recover S.

1. **Setup.** The trusted dealer T distributes a secret integer $S \geq 0$ to n participants.
   (a) T chooses a prime $p > \max(S, n)$, and set $a_0 = S$.
   (b) T defines the polynomial $f(x)$ over $\mathbb{Z}_p$ as $f(x) = \sum_{j=0}^{t-1} a_j x^j$, where $a_j \in_R \mathbb{Z}_p$ for $j \neq 0$ and $a_0 = S$.
   (c) T computes the n shares $s_i = f(i) \mod p$ for distinct $i \in \mathbb{Z}_p$ and securely transfers them to party p_i.
2. **Pooling of shares.** Any group of at least t participants pool their shares, providing enough distinct points to compute coefficients a_j by Lagrange interpolation. The secret is $f(0) = a_0$.

Starting from the t points, the secret $S = a_0$ is recovered using a formula that can be deduced from Lagrange interpolation:

$$S = \sum_{i=1}^{t} c_i s_i, \text{ where } c_i = \prod_{1 \leq j \leq t, j \neq i} \frac{s_j}{s_j - s_i}.$$

This scheme has the following properties:

1. *Perfect privacy* is preserved since given the knowledge of any $t-1$ or fewer shares, all values $0 \leq S \leq p-1$ of the shared secret remain equally probable.
2. It is *ideal*: the size of a share is equal to the size of the secret.
3. It is *extendable*: new shares can be computed and distributed without affecting existing shares. This means that the number of participants can be increased if needed.
4. It is possible to *vary the level of control* by providing multiple shares to one party.

5. The *security of the scheme is theoretical* and does not rely on a "hard problem" (like many asymmetric cryptographic algorithms).
6. It is a *linear scheme*.

The simplicity of its access structure mainly brings two disadvantages. First, participants are all equal in rights and it is not possible to privilege some over others. Yet in real life, there are often hierarchies between people. However, it is still possible to assign several shares to a single participant in order to increase its power. The second drawback comes from the density of the structure: any group with a size above the fixed threshold is qualified. This does not simplify the tracing to identify the group that has obtained the secret. When this tracing property is required, it may be a good idea to use a sparse access structure. Suppose we have a $(5, 2)$-access structure. Even if we have identified a participant who did not participate in the reconstruction of the secret, there are still six possible pairs. Now, if we consider a sparse access structure, for example $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, we know that if the participant 1 did not participate in the reconstruction of the secret, then participant 2 did it and he can possibly reveal the identity of the other participant.

Shamir scheme has many direct applications but also indirect applications as for example secure multiparty computation (SMPC) which allows parties to jointly compute a function over their inputs while keeping those inputs private. It is thus a question of protecting participants' privacy from each other. This problem is difficult and started to be studied in the 1970s [37]. Several protocols have been developed since then. For more about multiparty computation, one can consult for example [27]. The following example presents a very simple example of SMPC.

Example 1.5.1. Alice, Bob and Iwan want to know the total sum of their salaries without disclosing theirs. To do this, each participant constructs a polynomial (that is kept secret) of degree 2 with random coefficients except for the constant term which is equal to the participant's salary. Let P_1, P_2, P_3 be the respective polynomials of Alice, Bob and Iwan. Thus, $P_1(0)$ is equal to the salary of Alice, $P_2(0)$ is equal to the salary of Bob and so on.

Now Alice evaluates its polynomial at three defined values (say 1, 2, 3) i.e. $P_1(x)$ for $x = 1,\ 2,\ 3$, while Bob and Iwan evaluate their own polynomial at the same values. Then, Alice sends $P_1(2)$ to Bob and $P_1(3)$ to Iwan, Bob sends $P_2(1)$ to Alice and $P_2(3)$ to Iwan and Iwan sends $P_3(1)$ to Alice and $P_3(2)$ to Bob. At this stage, Alice knows

$$P_1(1),\ P_2(1),\ P_3(1),$$

Bob knows

$$P_1(2),\ P_2(2),\ P_3(2),$$

and Iwan knows

$$P_1(3),\ P_2(3),\ P_3(3).$$

This means that each participant knows the (public) value $P(i)$ with $P = P_1 + P_2 + P_3$ for respectively $i = 1,\ 2,\ 3$ by just adding their shares. The polynomial P of degree 2 can be constructed with these 3 values using an interpolation and $P(0)$ gives the sum of the three salaries. In fact, this protocol can also be used for different other applications like anonymous voting. The security of the scheme relies on the fact that P_i are kept secret and that each participant keeps one of its three evaluations secret. Moreover, the participants must execute the protocol correctly without cheating.

In the following sections, we will consider secret-sharing schemes that are constructed using linear codes. In particular, we will see that Shamir scheme can also be constructed from special linear codes called Reed Solomon codes.

1.6. Basics of Coding Theory

We briefly introduce the main notions of error correcting block codes. For more details about this topic, see [30], Let $\mathbb{F}_q$ be the finite field of order q, where q is a prime power. Any non-empty subset C of $\mathbb{F}_q^n$ is called a *code* and the parameter n is called the length of the code. Each vector in C is called a codeword of C. The *Hamming weight*

$\mathrm{wt}(v)$ of a vector v in $\mathbb{F}_q^n$ is the number of its non-zero coordinates. The *Hamming distance* between two vectors of $\mathbb{F}_q^n$ is the number of coordinates that differ between these two vectors. The *minimum distance* of a code C is the smallest of all Hamming distances between different codewords in C. It follows from this definition that a code with minimum distance $d_{\min}$ can correct $e := \lfloor (d_{\min} - 1)/2 \rfloor$ errors, since spheres of radius e are pairwise disjoint. The number e is called the *packing radius* or *error correction capability* of the code. If $d_{\min}$ is even the code can detect $d_{\min}/2$ errors, meaning that a received vector cannot have distance $d_{\min}/2$ to one codeword and distance less than $d_{\min}/2$ to another one. It may however have distance $d_{\min}/2$ to more codewords. Notice that for a subspace $V \subset \mathbb{F}_q^n$, the minimum distance of V can be computed by linearity as the minimum non-zero Hamming weight in V that is

$$d_{\min}(V) = \min\{\mathrm{wt}(v) | v \in V \setminus \{0\}\}.$$

The notion of error can be completed by the notion of *erasure*. We use the term erasure when there is an ambiguity on the value of a coordinate at a given position in the received vector. Thus erasures can be considered as errors in known positions. A code can decode errors and erasures simultaneously. If C is a code of length n with minimum distance $d_{\min}$ then it can correct b errors and c erasures as long as $2b + c < d_{\min}$. In other words, the transmitted codeword should be retrievable if during the transmission at most c of the symbols in the word are erased and at most b received symbols are incorrect.

Definition 1.6.1. An $[n, k, d]$ *linear code* $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$, where k is the dimension and $d = d_{\min}(\mathcal{C})$ is the minimum Hamming weight.

A *generator matrix* G for a code $\mathcal{C}$ is a matrix whose rows form a basis for $\mathcal{C}$. For any linear code $\mathcal{C}$, we denote by $\mathcal{C}^\perp$ its *dual* under the usual inner product. A code $\mathcal{C}$ is said to be *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$ and it is *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$. Whenever d is used to denote the minimum distance of $\mathcal{C}$, $d^\perp$ is used to denote the minimum distance of $\mathcal{C}^\perp$. The standard inner product of two vectors x and y is denoted $x \cdot y$.

Example 1.6.2. Let $\mathcal{C}$ be a $[5, 2, 2]$ linear binary code with generator matrix

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

This code is self-orthogonal since all its codewords are orthogonal to each other. It is not self-dual since the vector $v = (11100) \notin \mathcal{C}$ belongs to the dual code. Let d be the minimum distance of $\mathcal{C}$, then the fact that $\mathcal{C} \subseteq \mathcal{C}^{\perp}$ implies that $d^{\perp} \leq d$. It is easy to see that $d = 2$ and $d^{\perp} = 1$ (because the vector $(10000) \in \mathcal{C}^{\perp}$).

A linear complementary dual code also called *LCD* is such that $C \cap C^{\perp} = \{0\}$.

As seen before, the support of v is given by $\text{supp}(v) = \{i : v_i \neq 0, 1 \leq i \leq n\}$. A vector v_2 covers a vector v_1 if the support of v_2 contains that of v_1.

Definition 1.6.3. A non-zero codeword of C is called *minimal vector* if its support does not properly contain the support of another non-zero codeword.

The *covering problem* of a linear code is to determine the set of all its minimal vectors.

Definition 1.6.4. A linear code C is minimal if every non-zero codeword of C is a minimal vector.

Example 1.6.5. The binary Simplex code of parameters $[7, 3, 4]$ is minimal since it is a one weight code.

A linear code C can also be defined by its parity check matrix H. This is a matrix which describes the linear relations that the components of a codeword must satisfy. The matrix H is the generator matrix of the dual of C. This means that a codeword c is in C if and only if $cH^T = 0$. Thus, we have $HG^T = GH^T = 0$.

For any $[n, k, d]$-code, the following inequality known as the Singleton bound holds, $d \leq n - k + 1$. In case of equality, the code is called maximum distance separable (MDS). The most famous MDS codes are Reed–Solomon codes which are used in many applications,

and in particular in secret-sharing schemes. Some properties of MDS codes are recalled in the following lemma.

Lemma 1.6.6. *Let C be an $[n, k, d]$-code. Then, the following statements are equivalent*:

(1) *the minimum distance $d_{\min}$ of C is such that $d_{\min} = n - k + 1$;*
(2) *any k columns of a generator matrix of C are linearly independent*;
(3) *$C^{\perp}$ is an $[n, n - k, k + 1]$-MDS code.*

1.7. Code-Based Constructions of Secret-Sharing Schemes

McEliece and Sarwate [34] were the first to observe a connection between secret-sharing and error-correcting codes. Then some general relationships between linear codes and secret-sharing schemes were established by Massey [31] and Blakley and Kabatianskii [8].

In terms of vector spaces, Brickell [12] studied also this kind of secret-sharing schemes. A generalization of this approach was given by Bertilsson [6] and then by van Dijk [41].

Two approaches can be considered for the construction of secret-sharing schemes based on linear codes. These two constructions can be described as follows.

1.7.1. Construction 1

The first construction uses an $[n, k, d]$ linear code C over $\mathbb{F}_q$ ($k \geq 2$) with generator matrix $G = (g_0, \ldots, g_n)$. Recall that G is a $k \times n$ matrix, thus, g_i are vectors of length k. Let P_0 be the dealer and $P_i, 1 \leq i \leq n$, be the participants. The dealer chooses a vector $u = (s = u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ whose first coordinate is the secret s and the other coordinates are chosen randomly. He calculates the codeword $y = (s_0, \ldots, s_{n-1}) \in \mathbb{F}_q^n$ corresponding to this information vector u as $y = uG$. The dealer P_0 gives the share s_i to participant P_i. Notice that the vector of shares y is a linear combination of the rows of the generator matrix of the code G. We have

$$(s, u_1, \ldots, u_{k-1})G = (s_1, \ldots, s_n).$$

Moreover, it is easy to see that a set of shares $\{s_{i_1}, s_{i_2}, \ldots, s_{i_m}\}$ determines the secret s if and only if the vector $e_1 = (1, 0, \ldots, 0)^T$ is a linear combination of $g_{i_1}, g_{i_2}, \ldots, g_{i_m}$, where g_i is the ith column of G. Furthermore, the secret-sharing scheme is perfect and ideal.

The secret recovering procedure goes as follows. Given a set of shares $\{s_{i_1}, s_{i_2}, \ldots, s_{i_m}\}$ which can determine the secret, the group of m participants solves the equation

$$e_1 = \sum_{j=1}^{m} x_j g_{i_j}$$

to obtain the combination of the columns of G which is equal to e_1. Then the secret is

$$s = u \cdot e_1 = \sum_{j=1}^{m} x_j u \cdot g_{i_j} = \sum_{j=1}^{m} x_j s_{i_j}.$$

Thus, we can state the following theorem.

Theorem 1.7.1. *Let C be an $[n, k, d]$-code over $\mathbb{F}_q$ with generator matrix $G = [g_1, \ldots, g_n]$. The access structure of the secret-sharing scheme based on G with respect to the first construction is given by*

$$\mathcal{A} = \left\{ \{P_i \ : \ i \in \mathrm{supp}((x_1, \ldots, x_n))\} \ : \ e_1 = \sum_{j=1}^{m} x_j g_{i_j} \right\}.$$

This theorem shows that the access structure depends on the choice of the underlying generator matrix G.

Secret-sharing schemes based on MDS codes over $\mathbb{F}_q$ with respect to the first construction were investigated by Renvall and Ding in [35]. Their access structures are known and documented in the following theorems.

Theorem 1.7.2. *Let G be a generator matrix of an $[n, k, n-k+1]$-MDS code C defined over $\mathbb{F}_q$. In the secret-sharing scheme based on G with respect to the first construction, any k shares determine the secret.*

Theorem 1.7.3. *Let G be a generator matrix of an $[n, k, n-k+1]$-code C defined over $\mathbb{F}_q$ such that its ith column is a multiple of e_1.*

In the secret-sharing scheme based on G with respect to the first construction, a set of shares determines the secret if and only if it contains the ith share or its cardinality is no less than k.

Theorem 1.7.3 is interesting because it describes an access structure which can have some useful applications. For example, in a company, any group containing the director can obtain the secret. Moreover, if a group does not contain the director, it must contain at least k participants to obtain the secret. In fact it is always possible to form a generator matrix whose one column is e_1 by taking adequate linear combinations of some rows of the matrix.

Theorem 1.7.4. *Let G be a generator matrix of an $[n, k, n-k+1]$-code C over $\mathbb{F}_q$ such that e_1 is a linear combination of g_1 and g_2, but not a multiple of any of them, where g_i denotes the ith column vector of G. The access structure of the secret-sharing scheme based on G with respect to the first construction is then described as follows*:

(1) *any set of k shares determines the secret;*
(2) *a set of shares with cardinality $k-2$ or less determines the secret if and only if the set contains both s_1 and s_2;*
(3) *a set of $k-1$ shares $\{s_{i_1}, s_{i_2}, \ldots, s_{i_{k-1}}\}$*
   - *determines the secret when it contains both s_1 and s_2;*
   - *cannot determine the secret when it contains one and only one of s_1 and s_2;*
   - *determines the secret if and only if* $\mathrm{Rank}(e_1, g_{i_1}, \ldots, g_{i_{k-1}}) = k-1$ *when it contains none of s_1 and s_2.*

Theorem 1.7.5. *Let G be a generator matrix of an $[n, k, n-k+1]$-code C over $\mathbb{F}_q$. The secret-sharing scheme based on G with respect to the first construction is a (k, n)-threshold scheme if and only if e_1 is not a linear combination of any $k-1$ columns of G.*

The following theorem, also presented in [35], gives a relation between access structure and the minimum distance of the code when considering the first construction.

Theorem 1.7.6. *Let G be a generator matrix of an $[n, k, n-k+1]$-code C over $\mathbb{F}_q$. In the secret-sharing scheme based on G with respect to the first construction, if each of two sets of shares $\{s_{i_1}, \ldots, s_{i_l}\}$ and*

$\{t_{j_1}, \ldots, t_{j_m}\}$ determines the secret, but no subset of any of them can do so, then

$$|\{i_1, \ldots, i_l\} \cup \{j_1, \ldots, j_m\}| - |\{i_1, \ldots, i_l\} \cap \{j_1, \ldots, j_m\}| \geq d^{\perp},$$

where $d^{\perp}$ is the minimum distance of the dual code C.

The access structure of a secret-sharing scheme built using the first construction may be very complex. Few investigations have been made on the subject.

Shamir scheme: We already saw that Shamir secret-sharing scheme has originally been presented in terms of polynomials. We show that it is in fact a scheme based on a linear code which is MDS.

Consider the information vector (which is kept secret) $a = (a_0, a_1, \ldots, a_{k-1})$, where $a_0 = s$ and the other coordinates are chosen randomly in the considered field. This vector defines a polynomial $f(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$. The vector of shares is the codeword $c = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$, where all the α_i are distinct. Let G be the generator matrix of this $[n, k]$-code

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \alpha_3^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}.$$

We have

$$c = aG.$$

The matrix G is the transpose of a Vandermonde matrix whose determinant is known to be non-zero (since all α_i are distinct). Thus, any k columns in G are linearly independent. By Lemma 1.6.6, this means that the minimum distance of the code is $d = n - k + 1$ (i.e. the code is MDS) and the code can correct $d - 1 = n - k$ erasures. Thus participants are able to obtain the full codeword as soon as they know k coordinates.

We can also consider the $[n + 1, k]$ code of generator matrix $G' = (e_1, G)$, where $e_1 = (1, 0, \ldots, 0)^T$. This latter code is also

MDS with parameters $[n+1, k, d+1]$. The generated codeword is $(f(0), f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n))$, where $f(0)$ is the secret and the other coordinates are the shares. Now, if participants know k coordinates, they can retrieved the full codeword including the secret which is in the first coordinate.

In fact, we can notice that there is an equivalence between the existence of a (k, n)-linear threshold secret-sharing scheme and a $[n+1, k]$ MDS code.

Remark 1.7.7. The code used to construct the Shamir secret-sharing scheme is a Reed–Solomon code, as was observed in [34].

Blakley scheme: Blakley scheme can be presented in terms of hyperplane geometry: to implement a (t, n)-threshold scheme, each of the n users is given a hyper-plane equation in a t-dimensional space over a finite field such that each hyperplane passes through a certain point. The intersection point of the hyperplanes is the secret. The secret can be retrieved by any group of t participants.

An affine hyperplane in a t-dimensional space with coordinates in a field $\mathbb{F}_q$ can be described by a linear equation of the following general form:

$$a_1x_1 + a_2x_2 + \cdots + a_tx_t = b.$$

The intersection of any t of these hyperplanes gives the intersection point. The secret can be any of the coordinates of the intersection point or any function of the coordinates. We take the secret to be the first coordinate of the point of intersection.

Let $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be the secret point whose first coordinate is the secret (i.e. $x_1 = s$) and the other ones are random. The dealer construct randomly the $t \times n$ matrix G with coordinates $a_{ij} \in \mathbb{F}_q$. Then he considers the linear system $Gx = y$, where $y = (s_1, \ldots, s_n)$. The dealer sends to the ith participant the value s_i along with the ith row of G. In fact, the whole matrix G can also be made public once and for all. Any group $S = \{i_1, \ldots, i_t\}$ of t participants can retrieve the secret by forming the matrix G_S and solve the equation $G_Sx = s_S$, where s_S is the vector of shares of the t participants. The secret is the first coordinate of x. In order to get a solution G_S must be invertible which is the case if a Vandermonde matrix is chosen.

Blakley's scheme is less space-efficient than Shamir's; while Shamir's scheme is ideal, Blakley's shares are t times larger than the secret, where t is the threshold number of players. Blakley's scheme can be tightened by adding restrictions on which planes are usable as shares. The resulting scheme is equivalent to Shamir's polynomial system.

1.7.2. Construction 2

Secret-sharing schemes based on codes with respect to the second construction were considered by Massey. The two constructions may seem different but they are related. In the first approach all the shares form a complete codeword of the code, while in the second one all the shares form only part of a codeword. But as remarked by Van Dijk [42] we can switch from the second method to the first one since the generator matrix of the first method is obtained by puncturing the generator matrix of the second method.

In 1993, James Massey [31, 32] showed that a perfect and ideal secret-sharing scheme can be constructed using a linear code C. He introduced the notion of minimal codewords of a code and pointed out the relationship between access structures and minimal codewords of $C^{\perp}$.

The second construction uses an $[N = n + 1, k, d]$ linear code C over $\mathbb{F}_q$ (all the codes considered in this chapter are defined over $\mathbb{F}_q$). Let $G = (g_0, g_1, \ldots, g_n)$ be a generator matrix of C (it is a $k \times (n+1)$ matrix). The secret is denoted $s \in \mathbb{F}_q$ and the participants are $P_1,\ P_2, \ldots, P_n$.

The dealer computes the shares by randomly choosing a vector

$$u = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$$

such that

$$s = ug_0.$$

In practice, the dealer can randomly select $k - 1$ coordinates and then compute the kth in order to obtain the right property. Thus, there are q^{k-1} possible such u. This vector u is seen as an information vector and the dealer computes the codeword

$$y = uG = (s, s_1, \ldots, s_n).$$

The first coordinate of y is the secret s and the other coordinates are the shares corresponding to the secret.

Then the dealer P_0 gives the share s_i to participant P_i, $(1 \leq i \leq n)$.

The vector of shares y is a linear combination of the rows of G. Moreover, any group $S = \{P_{i_1}, \ldots, P_{i_m}\}$ of m participants can retrieve the secret if and only if g_0 is a linear combination of $g_{i_1}, \ldots, g_{i_m}$. Therefore, the access structure of the scheme is

$$\mathcal{A} = \left\{ \{P_{i_1}, \ldots, P_{i_m}\} : \begin{array}{c} 1 \leq i_1 < \cdots < i_m \leq n \text{ and} \\ g_0 \text{ is a linear combination of } g_{i_1}, \ldots, g_{i_m} \end{array} \right\}.$$

If g_0 is not a linear combination of $g_{i_1}, \ldots, g_{i_m}$, the set of shares $\{s_{i_1}, \ldots, s_{i_m}\}$ gives no information about s. Thus, the secret-sharing scheme is perfect.

Suppose m participants pool their shares $t_{i_1}, \ldots, t_{i_m}$. This set is qualified if and only if there exists in the dual code $\mathcal{C}^{\perp}$ a codeword

$$c^* = (1, 0, \ldots, 0, \lambda_{i_1}, 0, \ldots, 0, \lambda_{i_m}, 0, \ldots, 0)$$

of Hamming weight at least 2. Equivalently, we have

$$g_0 = -\sum_{j=1}^{m} \lambda_{i_j} g_{i_j}.$$

Since $s = ug_0$, the secret s can be computed as

$$s = ug_0 = -\sum_{j=1}^{m} \lambda_{i_j} u g_{i_j} = -\sum_{j=1}^{m} \lambda_{i_j} t_{i_j}.$$

The scheme being monotone, a group covering a qualified group is also qualified. Thus, the set of qualified groups can be deduced from the set of minimal qualified groups (which is a spanning set of the set of qualified groups). This means that in order to obtain the access structure it is sufficient to determine the set of minimal qualified groups.

In order to find minimal qualified groups, we introduce the notion of minimal codewords:

Definition 1.7.8. A codeword $c \in \mathcal{C}$ is called a *minimal codeword* if its first coordinate is 1 and covers no other codeword whose first coordinate is 1.

Remark 1.7.9. Notice that the notion of minimal codeword is more stringent than that of minimal vector.

Example 1.7.10. Consider again Example 1.6.2. The dual code $\mathcal{C}^{\perp}$ is a $[5, 3, 1]$ linear binary code with generator matrix

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

The codeword (11111) is not a minimal codeword since it covers 2 other codewords: (11100) and (10011). Indeed, the minimal codewords of $\mathcal{C}^{\perp}$ are (11100) and (10011).

In order to find the set of qualified groups, it is enough to find the set of minimal qualified groups. In fact, Massey shows in [31] that there is a one-to-one correspondence between the set of minimal qualified groups and the set of minimal codewords of $\mathcal{C}^{\perp}$. Therefore, searching for the access structure of the scheme means determining the minimal codewords of $\mathcal{C}^{\perp}$. In other words, the set of groups A such that

$$A = \{P_i : i \in \text{supp}(c) : c \text{ is a minimal codeword of } \mathcal{C}^{\perp}\}$$

is a spanning set for the set of qualified groups. We can see that, unlike the first construction, the access structure of the secret-sharing scheme is independent of the choice of the generator of the code.

Example 1.7.11. Let C be a $[6, 4, 2]$-linear code over $\mathbb{F}_5$ with generator matrix

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 & 4 & 1 \\ 1 & 0 & 1 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 & 0 & 4 \end{pmatrix}$$

and parity-check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 4 & 0 & 4 \end{pmatrix}.$$

Suppose the secret is $s = 3 \in \mathbb{F}_5$. The dealer chooses $u = (1123)$ and computes the codeword $t = uG = (312340)$. Thus, t_0 is the secret and the shares are $t_1 = 1,\ t_2 = 2,\ t_3 = 3,\ t_4 = 4,\ t_5 = 0$. The dealer then sends the share t_i to participant p_i. The set of codewords whose first coefficient is 1 is

$$\{(100111), (111010), (122414), (133313), (144212)\}$$

and the set of minimal codewords is

$$\{(100111), (111010)\}.$$

Therefore, the set $\{\{p_3, p_4, p_5\},\ \{p_1, p_2, p_4\}\}$ is a spanning set for the set of qualified groups.

Unfortunately, determining the minimal codewords is hard for general linear codes, which means that obtaining the access structures of a scheme based on general linear codes is also hard [19, 28]. Renvall and Ding [35] show that the minimum distance $d^\perp$ of $C^\perp$ gives a lower bound on the size of any minimal qualified group in the secret-sharing scheme based on C.

Theorem 1.7.12. *Let C be an $[n, k, d]$-code. In the secret-sharing scheme based on C with respect to the second construction, any set of $d^\perp - 2$ or fewer shares do not give any information on the secret, and there is at least one set of $d^\perp$ shares that determines the secret.*

The following theorem, due to Ding and Yuan in [19], describes the access structure of the secret-sharing scheme based on the dual of a minimal linear code.

Theorem 1.7.13. *Let C be an $[n + 1, k, d]$-code, and let $H = [h_0, h_1, \ldots, h_n]$ be its parity-check matrix. If $C^\perp$ is minimal, then in the secret-sharing scheme based on C with respect to the second construction, the set of participants is $P = \{P_1, P_2, \ldots, P_n\}$, and there are altogether q^{n-k} minimal qualified groups.*

- *When $d = 2$, the access structure is as follows. If h_i is a multiple of h_0, $1 \le i \le n$, then participant P_i must be in every minimal qualified group. If h_i is not a multiple of h_0, $1 \le i \le n$, then participant P_i must be in $(q-1)q^{n-k-1}$ out of q^{n-k} minimal qualified groups.*

- *When $d \geq 3$, for any fixed $1 \leq t \leq \min(n-k, d-2)$ every group of t participants is involved in $(q-1)^t q^{n-k-t}$ out of q^{n-k} minimal qualified groups.*

Example 1.7.14. Let C be the binary code of length $n+1 = 15$, dimension $k = 11$, minimum distance $d = 3$ and generator matrix

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

Its dual is generated by

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

and contains 16 codewords. This code is equivalent to the binary Hamming code of length 15. The $2^{n-k} = 8$ minimal qualified groups are

$$\{4,7,8,10,12,13,14\}, \quad \{1,5,8,9,11,13,14\}, \quad \{1,2,6,9,10,12,14\},$$
$$\{2,4,5,6,7,11,14\}, \quad \{2,3,4,5,9,12,13\}, \quad \{1,2,3,7,10,11,13\},$$
$$\{1,3,5,6,7,8,12\}, \quad \{3,4,6,8,9,10,11\}.$$

We can see that every participant is involved in four minimal qualified groups, every group of two participants is involved in two minimal qualified groups and every group of three participants is involved in one minimal qualified group.

Example 1.7.15. Let C be the binary cyclic code of length $n+1=9$, dimension $k=7$, minimum distance $d=2$ and generator matrix

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

Its dual is generated by

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

Since $h_3 = h_6 = h_0$, participants P_3 and P_6 are dictators: they are involved in all qualified groups.

The second construction has been studied in many different works. We can mention the work of Anderson, Ding, Helleseth and Kløve [2] which deals with codes having few words and the relations between t-designs and access structures. In 2005, Carlet, Ding and Yuan [15] used perfect nonlinear functions to construct several classes of linear codes and analyze the access structures of the secret-sharing schemes based on the dual of these codes. In 2008, Dougherty, Mesnager and Solé [20] considered self-dual codes to construct secret-sharing schemes. Secret-sharing schemes based on additive codes over $\mathbb{F}_4$ and their connections with t-designs were considered by Kim and Lee in [26].

1.8. Multisecret-Sharing Schemes

A *multisecret-sharing scheme* is a protocol to share m arbitrarily related secrets $s_1, \ldots, s_m$ among a set of n participants $\{P_1, \ldots, P_n\}$. It is an important variant of secret-sharing scheme which arises when multiple secrets must be distributed with different thresholds to the participants [4, 11, 40, 43]. Jackson *et al.* [24] studied the case where each secret s_i is associated with a threshold t_i. In this scheme, called

multisecret threshold schemes, only a group of more than t_i participants can recover s_i. Blundo *et al.* [11] have shown that in order to obtain unconditional security for multisecret threshold schemes, the size of each share must be, at least, linear in the number of secrets. These lower bounds have been supplemented by more recent work from Masucci [33] who presented a weaker notion of security for multisecret sharing schemes in unconditional settings, and gave some lower bounds for the size of each share. Herranz *et al.* [22] then proved that each share in multisecret threshold schemes must be linear in the number of secrets. Multisecret sharing scheme is always very studied as shown by very recent articles on the subject (see for example [1, 13, 14]).

We assume that each element of the secret space S_i is equally likely to be the ith secret for each i such that $1 \leq i \leq k$. The share space is denoted T_i for $1 \leq i \leq n$. We consider the case where both spaces are equal to $\mathbb{F}_q$. Let $S = S_1 \times \cdots \times S_k$ and $T = T_1 \times \cdots \times T_n$ be product spaces.

Definition 1.8.1. Let $s = (s_1, \ldots, s_k)$ be the vector consisting of k secrets and $t = (t_1, \ldots, t_n)$ be the set of the shares. A (k, m, n)-multisecret sharing scheme consists of two protocols, Share and Reconstruct.

The Share function

$$\begin{aligned} f: \ & S \to T \\ & f(s) = t \end{aligned}$$

is a function that constructs the n shares from the k secrets. Share t_i is given to the ith participant.

The Reconstruction phase consists in recovering the secret from the shares. In fact, any group of m shares is able to reconstruct secret s_j but no group of strictly less than m shares is able to reconstruct the secret.

A multisecret-sharing scheme is said to be *linear* if for all $a, a' \in \mathbb{F}_q$ and all $s, s' \in S$, $f(as + a's') = af(s) + a'f(s')$. The scheme is called *affine* if there is a constant λ such that $f - \lambda$ is linear. In fact, if the scheme is linear, it must be of the form $f(s) = sG$, where G is $k \times n$ matrix with rank k. Then whenever we define such a scheme, we define an $[n, k, d]$-code over $\mathbb{F}_q$ and the Share function is just an encoding mapping of a linear code.

For a group of m participants, recovering the secret is equivalent to solving a linear equation. Suppose the group knows the shares $t_{i_1}, \ldots, t_{i_m}$ $(m \leq n)$ and let G' be a submatrix of G consisting of the t_{i_1}th, $\ldots, t_{i_m}$th columns of G. The linear equation is

$$sG' = (t_{i_1}, \ldots, t_{i_m}) \tag{1.8.1}$$

and solving this equation is not difficult, for example using Gaussian elimination method.

We consider here the particular case $m = k$ which is adapted to construct linear multisecret-sharing schemes based on MDS codes.

Theorem 1.8.2. *A multisecret-sharing scheme is a $[k, k, n]$-threshold scheme if and only if*

(1) *the linear code C with generator matrix G is MDS and*
(2) *any set of $k-1$ column vectors of G generates a $[k, k-1, 2]$-MDS code.*

This means that in order to obtain a $[k, k, n]$-threshold scheme, we need a $[n, k, n-k+1]$-code and k codes with parameters $[k, k-1, 2]$.

In a $[k, k, n]$-multisecret-sharing scheme, each share contains the same amount of information about the multisecret, and two groups of shares give the same amount of information about the multisecret if and only if these two groups have the same number of shares.

We can use a Reed–Solomon code to construct such a scheme. Let $\alpha = (\alpha_1, \ldots, \alpha_n)$, where α_i are distinct element of $\mathbb{F}_q$. It is well known that the following generator matrix:

$$G = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \alpha_3^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}$$

generates an MDS $[n, k, n-k+1]$-code. In order that condition (2) of Theorem 1.8.2 holds, we have to choose α_i such that: any $(k-1) \times (k-1)$ submatrix of G has rank $k-1$ if and only if for any set

of indices $1 \leq i_1 < \cdots < i_{k-1} \leq n$ we have

$$\sum_{1 \leq u_1 < \cdots < u_j \leq k-1} \alpha_{i_{u_1}} \alpha_{i_{u_2}} \cdots \alpha_{i_{u_j}} \neq 0 \quad \text{for all } j = 1, 2, \ldots, k-2.$$

Then using Eq. (1.8.1), we can obtain the vector of shares.

Example 1.8.3. We consider the code of parameters $[5, 3, 3]$ over F_{13} with generator matrix

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 3 \\ 1 & 4 & 3 & 12 & 9 \end{pmatrix}.$$

In this matrix, condition (2) of Theorem 1.8.2 is met. Therefore, G provides a $[3, 3, 5]$-multisecret threshold scheme.

References

[1] A. Alahmadi, A. Altassan, A. AlKenani, S. Çalkavur, H. Shoaib and P. Solé. A multisecret-sharing scheme based on LCD codes. *Mathematics*, 8:272, 2020, doi:10.3390/math8020272.

[2] R.J. Anderson, C. Ding, T. Helleseth and T. Kløve. How to build robust shared control systems. *Des. Codes Cryptography*, 15(2):111–124, 1998.

[3] A. Beimel. Secret-sharing schemes: a survey. In Chee Y.M. *et al.*, editor, *Coding and Cryptology. IWCC 2011*. Lecture Notes in Computer Science, Vol. 6639. Springer, Berlin, https://doi.org/10.1007/978-3-642-20901-7_2.

[4] A. Beimel, A. Ben-Efraim, C. Padro and I. Tyomkin. Multilinear secret-sharing schemes. In *Theory of Cryptography-TCC 2014*, pp. 394–418, 2014.

[5] M. Ben-Or, S. Goldwasser and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC'88*, pp. 1–10, 1988.

[6] M. Bertilsson, Linear codes and secret sharing, PhD thesis, Linköping University, 1993.

[7] G. Blakley. Safeguarding cryptographic keys. *AFIPS Conf. Proc.*, 48:313–317, 1979.

[8] G. Blakley and G. Kabatianskii. *Linear algebra approach to secret sharing schemes*. Selected Papers from the Workshop on Information Protection, Error Control, Cryptology, and Speech Compression, Springer-Verlag, Berlin, Heidelberg, December 1993.

[9] K. Bozkurt, K. Kaya, A. Selçuk and A. Güloğlu. Threshold cryptography based on blakley secret sharing, *ICS*, 2008.
[10] C. Blundo, A. De Santis and U. Vaccaro. On secret sharing schemes. *Inform. Process. Lett.*, 65(1):2532, 1998.
[11] C. Blundo, A.D. Santis, G.D. Crescenzo, A.G. Gaggia and U. Vaccaro. Multi-secret sharing schemes. In *CRYPTO '94*, pp. 150–163, 1994.
[12] E. Brickell. Some ideal secret sharing schemes. In Quisquater J.J. and Vandewalle J. editors, *Advances in Cryptology, EUROCRYPT '89. EUROCRYPT 1989*. Lecture Notes in Computer Science, Vol. 434. Springer, Berlin, 1989, https://doi.org/10.1007/3-540-46885-4_45.
[13] S. Çalkavur and P. Solé. Multisecret-sharing schemes and bounded distance decoding of linear codes. *Int. J. Comput. Math.*, 94(1):107–114, 2017.
[14] S. Çalkavur and P. Solé. Some multisecret-sharing schemes over finite fields. *Mathematics*, 8:654, 2020, doi:10.3390/math8050654.
[15] C. Carlet, C. Ding and J. Yuan. Linear codes from perfect nonlinear mappings and their secret sharing schemes. *IEEE Trans. Inform. Theory*, 51(6):2089–2102, 2005.
[16] D. Chaum, C. Crepeau and I. Damgard. Multi-party unconditionally secure protokols. In *STOC'88*, pp. 11–19, 1988.
[17] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *FOCS85*, pp. 383–395, 1985. doi:10.1109/SFCS.1985.64.
[18] R. Cramer, I.B. Damgard and J.B. Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015, https://doi.org/10.1017/CBO9781107337756.
[19] C.S. Ding and J. Yuan. Covering and secret sharing with linear codes. In *Discrete Mathematics and Theoretical Computer Science*, Lecture Notes in Computer Science, Vol. 2731. Springer-Verlag, pp. 11–25, 2003.
[20] S.T. Dougherty, S. Mesnager and P. Solé. Secret-sharing schemes based on self-dual codes. In *Proc. 2008 Information Theory Workshop*, pp. 338–342. IEEE Press, 2008.
[21] S. Fehr and U. Maurer. Linear VSS and distributed commitments based on secret sharing and pairwise checks. In *CRYPTO'02*, Lecture Notes in Computer Science, Vol. 2442, pp. 565–580, 2002.
[22] J.J. Herranz, A. Ruiz and G. Saez. New results and applications for multi-secret sharing schemes. *Des. Codes Cryptography*, 73(3):841–864, 2014.
[23] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Proc. of ISTCS'97*, IEEE Computer Society, pp. 174–184, 1997.

[24] W.-A. Jackson, K.M. Martin and C.M. O'Keefe. Multisecret threshold schemes. In *CRYPTO 1993*, pp. 126–135, 1993.

[25] E.D. Karnin, J.W. Greene and M.E. Hellman. On secret sharing systems. *IEEE Trans. Inform. Theory*, 29(1):3541, 1983.

[26] J.-L. Kim and N. Lee. Secret sharing schemes based on additive codes over GF(4). *Appl. Algebra Eng. Commun. Comput.*, 28(1):79–97, 2017.

[27] Y. Lindell. Secure multiparty computation (MPC), Cryptology ePrint Archive, Report 2020/300, 2020, https://eprint.iacr.org/2020/300.

[28] Z.H. Li, T. Xue and H. Lai. Secret sharing schemes from binary linear codes. *Inform. Sci.*, 180:4412–4419, 2011.

[29] J. Liu, S. Mesnager and L. Chen. Secret sharing schemes with general access structures. In Lin, D., Wang, X., Yung, M., editors, Information Security and Cryptology. Inscrypt 2015. Lecture Notes in Computer Science, Vol. 9589. Springer, Cham, 2016.

[30] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error Correcting Codes*. North-Holland, Amsterdam, 1977.

[31] J.L. Massey. Minimal codewords and secret sharing. In *Sixth Joint Swedish-Russian Workshop on Information Theory*, pp. 276–279, 1993.

[32] J.L. Massey. Some applications of coding theory in cryptography. In *Cryptography and Coding IV*, Formara Ltd, England, 1995.

[33] B. Masucci. Sharing multiple secrets: Models, schemes and analysis. *Des. Codes Cryptography*, 39(1):89–111, 2006.

[34] R.J. McEliece and D.V. Sarwate. On sharing secrets and Reed-Solomon codes. *Comm. ACM*, 24(9):583–584, 1981.

[35] A. Renvall and C. Ding. The access structure of some secret sharing schemes. In *Information Security and Privacy*, Lecture Notes in Computer Science, Vol. 1172, pp. 67–78. Springer, 1993.

[36] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, 1979.

[37] A. Shamir, R. Rivest and L. Adleman. "Mental Poker", Technical Report LCS/TR-125, Massachusetts Institute of Technology, April 1979.

[38] G.J. Simmons. An introduction to shared secret and/or shared control schemes and their application. In *Contemporary Cryptology*, IEEE Press, New York, pp. 441–497.

[39] Y. Song and Z. Li. Secret sharing with a class of minimal linear codes. Preprint arXiv 1202.4058, 2012.

[40] C. Tartary, J. Pieprzyk and H. Wang. Verifiable multi-secret sharing schemes for multiple threshold access structures. In *Information Security and Cryptology, Third SKLOIS Conf., Inscrypt 2007*, Xining, China, August 31–September 5, 2007, Revised Selected Papers, pp. 167–181, 2007.

[41] M. van Dijk. A linear construction of secret sharing schemes. *Des. Codes Cryptography*, 12:161–201, 1997. https://doi.org/10.1023/A:1008259214236.
[42] M. van Dijk. *Secret key sharing and secret key generation*, Ph.D. thesis, TU Eindhoven, 1997.
[43] M. van Dijk, W.-A. Jackson and K.M. Martin. A general decomposition construction for incomplete secret sharing schemes. *Des. Codes Cryptography*, 3:301–321, 1998.
[44] D. Welsh. *Matroid Theory*. Academic Press, London, 1976.
[45] A. Yao. Protocols for secure computations. In *FOCS. 23rd Annual Symp. Foundations of Computer Science* (*FOCS 1982*), pp. 160–164. doi:10.1109/SFCS.1982.88.