Secret-Sharing Schemes for General Access Structures: An Introduction[1]

Amos Beimel
Department of Computer Science
Ben-Gurion University of the Negev
Beer-Sheva, Israel.
E-mail: amos.beimel@gmail.com

March 19, 2025

[1]This monograph is dedicated to the memory of my Ph.D. advisor Benny Chor (1956–2021). I would like to thank Benny for introducing me to the field of secret sharing, guiding me in the early stages of my career, and trying to teach me how to "think".

Abstract

A secret-sharing scheme is a method by which a dealer distributes shares to parties such that only authorized subsets of parties can reconstruct the secret. Secret-sharing schemes are an important tool in cryptography and they are used as a building block in many secure protocols, e.g., secure multiparty computation protocols for arbitrary functionalities, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, and weighted cryptography (e.g., stake-based blockchains). The collection of authorized sets that should be able to reconstruct the secret is called an access structure. The main goal in secret sharing is to minimize the share size in a scheme realizing an access structure. In most of this monograph, we will consider secret-sharing schemes with information-theoretic security, i.e., schemes in which unauthorized sets cannot deduce any information on the secret even when the set has unbounded computational power. Although research on secret-sharing schemes has been conducted for nearly 40 years, we still do not know what the optimal share size required to realize an arbitrary n-party access structure is; there is an exponential gap between the best known upper bounds and the best known lower bounds on the share size.

In this monograph, we review the most important topics on secret sharing. We start by discussing threshold secret-sharing schemes in which the authorized sets are all sets whose size is at least some threshold t; these are the most useful secret-sharing schemes. We then describe efficient constructions of secret-sharing schemes for general access structures; in particular, we describe constructions of linear secret-sharing schemes from monotone formulas and monotone span programs and provide a simple construction for arbitrary n-party access structures with share size 2^{cn} for some constant $c < 1$. To demonstrate the importance of secret-sharing schemes, we show how they are used to construct secure multi-party computation protocols for arbitrary functions. We next discuss the main problem with known secret-sharing schemes – the large share size, which is exponential in the number of parties. We present the known lower bounds on the share size. These lower bounds are fairly weak, and there is a big gap between the lower and upper bounds. For linear secret-sharing schemes, which are a class of schemes based on linear algebra that contains most known schemes, exponential lower bounds on the share size are known. We then turn to study ideal secret-sharing schemes in which the share size of each party is the same as the size of the secret; these schemes are the most efficient secret-sharing schemes. We describe a characterization of the access structures that have ideal schemes via matroids. Finally, we discuss computational secret-sharing schemes, i.e., secret-sharing schemes that are secure only against polynomial-time adversaries. We show computational schemes for monotone and non-monotone *circuits*; these constructions are more efficient than the best known schemes with information-theoretic security.

The goal of this monograph is to present the known results on secret-sharing schemes, in particular, on secret-sharing realizing general access structures. We will review older results and the most advanced results; due to space constraints and trying to keep the monograph coherent, some advanced constructions

and proofs are omitted. The study of secret-sharing schemes uses tools from cryptography, complexity, and information theory; some relevant background from these areas is presented in the appendices.

Contents

List of Figures

Chapter 1

Introduction

A secret-sharing scheme is a tool used in many cryptographic protocols to process sensitive information. It involves a dealer who has a secret string (aka the secret), a set of n parties, and a collection Γ of (authorized) subsets of parties called the access structure. A secret-sharing scheme for Γ is a method by which the dealer distributes strings (called shares) to the parties such that: (1) any subset in Γ can reconstruct the secret from its shares, and (2) any subset not in Γ cannot reveal any partial information on the secret. The main goal in secret-sharing schemes is to minimize the share size (i.e., the number of bits in the strings representing the shares).

We start with a simple motivating example. Assume that in a bank there is a manager, 3 deputy managers, and 10 tellers; there is a safe in the branch and due to security considerations it can only be opened by enough trusted entities, i.e., it can be opened by (1) the manager, (2) two deputy managers, or (3) a deputy manager and 3 tellers. To enable such a policy we can employ a secret-sharing scheme to share the passcode of the safe. To clarify the notion of secret sharing, we next describe a simple secret-sharing scheme realizing a simple access structure.

Example 1.1 (An n-out-of-n Secret-Sharing Scheme)**.** Assume that there are n parties $p_1, \ldots, p_n$ and we require that all parties together can reconstruct the secret, while every subset of the parties gets no information on the secret, that is, the access structure is $\Gamma = \{\{p_1, \ldots, p_n\}\}$. To share a secret $s \in \{0,1\}$, the dealer chooses $n-1$ independent and uniformly distributed random bits $r_1, \ldots, r_{n-1}$ and computes $r_n \leftarrow s \oplus r_1 \oplus \cdots \oplus r_{n-1}$. The share of party p_i is r_i. The n parties can reconstruct the secret by computing the exclusive-or of their bits. On the other hand, every subset of $n-1$ bits is uniformly distributed, hence does not disclose any information on the secret.

Originally motivated by the problem of secure information storage, secret-sharing schemes have found numerous other applications in cryptography, distributed computing, and complexity theory, e.g., Byzantine agreement [151], secure multiparty computations [97, 34, 57, 64], threshold cryptography [75], access control [140], attribute-based encryption [101, 181, 182, 10], generalized oblivious transfer [164, 173], weighted cryptography (e.g., stake-based blockchains) [92, 38, 74, 176], and proving NP-hardness of the partial minimum circuit size problem [105]. We next describe such an application.

Example 1.2 (Attribute Based Encryption)**.** Public-key encryption is a powerful mechanism for protecting the confidentiality of stored and transmitted information. Nowadays, in many applications there is a provider that wants to share data according to some policy based on the user's credentials. In an attribute-based encryption system, presented by Sahai and Waters [160], each user has a set of attributes (i.e., credentials), and the provider will grant permission to decrypt the message if some predicate of the attributes holds (e.g., a user can decode an e-mail if she is a "FRIEND" and "IMPORTANT"). In [101, 181], it is shown that if the predicate can be described by an access structure that can be implemented by an efficient linear secret-sharing scheme, then there is an efficient attribute-based encryption system for this predicate.

In most of this monograph, we will study secret-sharing schemes with information-theoretic security, i.e., secret-sharing schemes in which an unauthorized set of parties cannot learn any information about the secret even if they are unbounded. We next provide a short overview and the known results about information-theoretic secret-sharing schemes.

Threshold secret-sharing schemes, where the subsets that can reconstruct the secret are all the sets whose cardinality is at least a certain threshold t, were introduced by Shamir [163] and Blakley [40]. Threshold secret-sharing schemes are the most used secret-sharing schemes and can be realized with short shares. i.e., the size of the shares for sharing an ℓ-bit secret is ℓ, provided that $\ell \geq \log(n)$ (where n is the number of parties).

Secret-sharing schemes for general access structures were introduced and constructed by Ito, Saito, and Nishizeki [107]. More efficient schemes were presented in, e.g., [36, 166, 49, 111, 39, 41, 76, 123, 6, 8, 9]. Some access structures can be realized by schemes with short shares, e.g., such schemes were presented by Benaloh and Leichter [36] for access structures that can be represented by small monotone formulas.

For arbitrary access structures, the share size in the best known secret-sharing schemes is exponential in the number of parties [107, 36, 123, 6, 8, 9], i.e., it is $(3/2)^n = 2^{0.585n}$ for an n-party access structure [9]. The best known lower bound is far from the above upper bounds; Csirmaz [66, 67] proved that for every n, there is an n-party access structure such that in any secret-sharing realizing it, the share size of at least one party is $\Omega(n/\log(n))$ and the total share size is $\Omega(n^2/\log(n))$. The known upper and lower bounds for secret-sharing schemes are summarized in Table 1.1. As discussed, we do not know what the optimal share size for realizing an arbitrary access structure is; it can be anywhere between exponential and quadratic.

Question 1.3. *What is the share size required for realizing an arbitrary n-party access structure?*[1]

Answering this open problem is an intriguing open problem. We do not even know the answer for natural access structures (e.g., access structures that can be represented by monotone circuits). Secret-sharing schemes are a relatively simple cryptographic primitive. Understanding their power and limitations is a fundamental question; it also may be the first step in understanding other (possibly more complex) protocols

[1] In the earlier version of this monograph [15], the author conjectured that exponential size shares are necessary. In light of the improvement in the share size of general access structures [123, 6, 8, 9] and the related question of message size for CDS protocols [124, 125], the author makes no conjectures on the share size.

	Share size	
Upper bound for threshold secret sharing	$\max\{\ell, \log(n)\}$	[40, 163]
Lower bound for threshold secret sharing	$\max\{\ell, \log(n) - 1\}$	[112, 114, 47]
Upper bound for arbitrary access structures	$2^{0.585n+o(n)} \cdot \ell$	[9]
Best known lower bound for an n-party access structure	$\Omega\left(\frac{n}{\log(n)} \cdot \ell\right)$	[67]

Table 1.1: Summary of the known results on information-theoretic secret-sharing schemes for sharing an ℓ-bit secret among n parties.

with information-theoretic security, e.g., secure multiparty protocols, private simultaneous messages (PSM) protocols, and private information retrieval (PIR) protocols.

In Chapter 9, we will discuss computational secret-sharing schemes, i.e., schemes in which all parties run in polynomial time and the security requires that an unauthorized set of parties running in polynomial time cannot learn any information on the secret. The schemes with computational security have smaller share size compared to the secret-sharing schemes with information-theoretic security. However, the secret-sharing schemes with computational security rely on currently unproven assumptions (e.g., the existence of one-way functions) compared to schemes with information-theoretic security whose security is proven without any assumptions.

1.1 Detailed Discussion on the Topics Covered in This Monograph

Threshold Secret Sharing. The most useful secret-sharing schemes are threshold secret-sharing schemes, introduced by Blakley [40] and Shamir [163] (Shamir's paper was cited more than 20,000 times). Threshold secret-sharing schemes were used in constructions of secure multiparty computations [97, 34, 57] and threshold cryptography [75]. In a t-out-of-n threshold secret-sharing scheme, a set of parties of size at least t can reconstruct the secret, while sets of size at most $t-1$ should gain no information on the secret (where n is the number of parties). For $t = 1, n$, there are t-out-of-n secret-sharing schemes in which the secret and each share are one bit (for $t = n$, see Example 1.1). Blakley [40] and Shamir [163] constructed t-out-of-n threshold secret-sharing schemes for an ℓ-bit secret and $2 \leq t \leq n-1$, where the size of each share is

$\max\{\log(n+1), \ell\}$. We describe Shamir's scheme [163] in Chapter 2. In every secret-sharing scheme, the share size is at least the size of the secret [112] (see proof in Chapter 7). Furthermore, for $2 \leq t \leq n-1$, the share size of at least one party in a t-out-of-n secret-sharing scheme is at least $\log(n) - 1$ [114, 47] (see also [54]). Thus, for threshold secret-sharing schemes we know the optimal share size. We provide the proof from [114] of a lower bound of $\log(n-t+2)$ in Section 2.3.

To bypass the above lower bounds, Blakley and Meadows [42] suggested a relaxation of threshold secret-sharing schemes; they defined (b,t)-ramp secret-sharing schemes, where $b < t$, in which every set of size at least t can reconstruct the secret, while every set of size at most b should not learn any information on the secret. For long enough secrets (namely, at least $2\log(n)$ bits), Blakley and Meadows constructed (b,t)-ramp secret-sharing schemes with share size $1/(t-b)$ times the size of the secret; this scheme is described in Section 2.4. Chen and Cramer [58] and Chen et al. [59] showed that when the gap is large, namely $t - b = O(n)$, the share size in (b,t)-ramp secret-sharing schemes can be reduced to $O(1)$.

Defining Secret-Sharing Schemes. The security of secret-sharing schemes asserts that unauthorized sets of parties should learn no information on the secret. There are (at least) two ways to formulate this requirement for schemes with information-theoretic security. The first way is to say that the probability distribution of the shares of an unauthorized set is the same for every two secrets. The second approach is to assume some probability distribution on the secrets and require that the posterior probability distribution on the secrets given the shares of an unauthorized set is equivalent to the prior probability distribution on the secrets. The latter definition requires knowing the prior probability distribution on the secrets while designing the scheme, which might be problematic. Nevertheless, we prove in Chapter 3 that the two definitions are equivalent for perfect secret-sharing schemes (i.e., schemes in which authorized sets learn absolutely no information on the secret). In particular, this implies that if a perfect secret-sharing scheme is secure under the second definition for one distribution on the secrets, then it is secure for every distribution with the same support; this result was originally proved by Blundo et al. [45]. We will mainly use the first definition to prove the security of secret-sharing schemes and use the second definition to prove lower bounds on the share size. We also define schemes with statistical security using the first definition, namely, we require that the probability distribution of the shares of an unauthorized set is nearly the same for every two secrets (i.e., the statistical distance between the distributions for every two secrets is negligible).

Efficient Secret-Sharing Schemes. In Chapter 4, we describe efficient constructions of secret-sharing schemes, i.e., constructions of schemes in which the share size is polynomial in the number of parties. These constructions follow the same paradigm: Use some small representation of an access structure and show a secret-sharing scheme whose share size is polynomial in the size of the representation. That is, Ito, Saito, and Nishizeki [107] showed how to realize an access structure represented by CNF and DNF formulas, Benaloh and Leichter [36] showed how to realize an access structure represented by *monotone formulas*, and Karchmer and Wigderson [111] showed how to realize an access structure represented by *monotone span program (MSPs)* (a special case of this construction appeared before in [49]). Monotone span programs

are a linear-algebraic computational model originally defined to prove lower bounds on the size of counting branching programs.

Secret-sharing schemes constructed from monotone span programs are equivalent to linear secret-sharing schemes [111, 14]; in a linear scheme, the secret is viewed as an element of a finite field, and the shares are obtained by applying a linear mapping to the secret and several independent random field elements. For example, the schemes of [163, 40, 107, 36, 166, 111] are all linear. For many applications, the linearity is important, e.g., for secure multiparty computation (MPC) protocols [64] and attribute-based encryption [182, 10]. In Section 4.5, we discuss useful properties of linear secret-sharing schemes that make them attractive, e.g., they are homomorphic – if we share two secrets and each party locally sums its shares, then we get shares of the sum of the secrets.

In Section 4.6, we present multilinear secret-sharing schemes [39, 41, 76], i.e., a linear secret-sharing scheme in which the secret is composed of more than one field element. Such schemes are provably more efficient than linear schemes [167, 5, 4, 19]. Specifically, Applebaum and Arkis [4] constructed multilinear secret-sharing schemes in which the size of the shares is 4 times the size of the secret for a family of $2^{2^{n/2}}$ access structures (alas for long secrets of size $2^{n^{n/2}}$); by counting arguments, the size of the shares in linear secret-sharing schemes for this family is exponential. It is important to note that for some applications, it is not known how to replace linear secret-sharing schemes with multilinear schemes, e.g., in the secure multiparty computation protocols of [64] that are secure against arbitrary Q^2 adversary structures.

Secret-Sharing Schemes with Share Size 2^{cn} for a Constant $c < 1$. Ito et al. [107] constructed the first secret-sharing scheme for general n-party access structures; the share size in their schemes is 2^n. For more than 30 years, no scheme with a share size better than $2^{n-o(n)}$ was known. Liu and Vaikuntanathan [123], in a breakthrough paper, constructed for every access structure a secret-sharing scheme with a share size $2^{0.994n}$. This was improved in a sequence of works [6, 8, 9], where the share size of the best known scheme is $(3/2)^{(1+o(1))n} < 2^{0.585n}$ [9]. These schemes rely on conditional disclosure of secret protocols, constructed by [124, 125]. In Chapter 5, we describe a fairly simple secret-sharing scheme for an arbitrary access structure with share size 2^{cn} for some constant $c < 1$. The idea of the scheme is to reduce the question of realizing an arbitrary n-party access structure to realizing a graph access structure, i.e., an access structure in which the minimal authorized sets are of size 2. Specifically, the graph access structure has $O(2^{n/2})$ parties and we only need a t-robust scheme for this graph for some $t \ll 2^{n/2}$, i.e., we allow sets of size greater than t to learn information on the secret. The construction presented in this monograph is not the best scheme known to date; however, it conveys a lot of the ideas of the best known constructions.

Secure Multiparty Computation Protocols from Secret Sharing. To demonstrate applications of secret-sharing schemes, we describe in Chapter 6 a secure multi-party protocol for an arbitrary function of [34] that is secure against semi-honest parties. That is, we consider a scenario in which the parties follow the protocol. However, at the end of the protocol, a set of less than half of the parties might try to learn additional information on the inputs of the other parties; a protocol is secure if such a set learns no information. The

protocol that we present is based on Shamir's threshold secret-sharing schemes, using its homomorphic property. The security of the protocol is perfect, i.e., the semi-honest parties learn absolutely no information that is not implied by their inputs and the output (even if they have unbounded power). Chapter 6 is less formal than the rest of this monograph and its purpose is to convey ideas on using secret-sharing schemes in general protocols.

Lower Bounds on the Share Size. The known lower bounds on the shares' size for sharing a secret realizing an arbitrary access structure are far from the above upper bounds. The best lower bound was proved by Csirmaz [66, 67], proving that, for every n, there is an n-party access structure such that sharing ℓ-bit secrets requires that the size of the share of at least one party is $\Omega(\ell n/\log(n))$ and the total share size is $\Omega(\ell n^2/\log(n))$. In Chapter 7, we provide this proof. This is an elegant proof that uses the entropy function and its properties. As discussed above, this lower bound is far from the best known upper bounds on the share size. Closing the exponential gap between these bounds is a fundamental open problem.

Lower Bounds on the Share Size of Linear Secret-Sharing Schemes. Many known secret-sharing schemes are *linear*. The best known linear secret-sharing schemes realizing arbitrary access structures have share size $2^{0.7563n}$ [123, 6, 8, 9, 2]. As discussed above, linear secret-sharing schemes are equivalent to monotone span programs. Lower bounds for monotone span programs and, therefore, for linear secret-sharing schemes were proved in [23, 11, 159, 89, 90, 158, 149, 150, 19].[2] In particular, exponential lower bounds for linear schemes for explicit access structures were given in [158, 149, 150]. Better exponential lower bounds for implicit access structures were proven in [11, 159, 19]. In Section 7.5, we describe a lower bound of $\Omega(2^{0.5n-o(n)})$ for a one-bit secret for almost all access structures [11] and in Section 7.6, we provide a proof of a lower bound of $n^{\Omega(\log(n)}\ell$ for an ℓ-bit secret for an explicit access structure [90]. The known upper and lower bounds for linear secret-sharing schemes are summarized in Table 1.2.

Ideal Secret Sharing. In every secret-sharing scheme, the share size of every non-redundant party (i.e., a party that participates in at least one minimal authorized set) is at least the size of the secret [112]. An access structure is ideal if it has a secret-sharing scheme in which the share size of each party is the size of the secret for some finite size of secrets (such a scheme is called ideal). That is, an access structure is ideal if it has the best possible share size. The strict requirement on the share size in ideal secret-sharing schemes enables us to analyze ideal access structures.

Ideal secret-sharing schemes and ideal access structures have been studied in many papers, e.g. [49, 165, 50, 162, 168, 53, 18, 127, 132, 44, 108, 99, 167, 133, 137, 146, 172, 174, 29, 128, 83, 85, 17, 110]. Brickell and Davenport [50] have shown an interesting connection between ideal access structures and matroids, combinatorial structures, defined by Whitney in 1935 [184], that abstract and generalize linear spaces and cycles

[2] The super-polynomial lower bounds of [90] also hold for multilinear secret-sharing schemes (where the secret can be composed of more many field elements [17]. We remark that applications that require linear secret-sharing schemes cannot necessary use multilinear secret-sharing schemes.

	Share size of linear schemes	
Upper bound for threshold secret sharing	$\max\{\ell, \log(n)\}$	[40, 163]
Lower bound for threshold secret sharing	$\max\{\ell, \log(n) - 1\}$	[112, 114, 111]
Upper bound for arbitrary access structures	$2^{0.7563n+o(n)} \cdot \ell$	[2]
Best known lower bound for an n-party access structure	$\Omega\left(\max\left\{2^{n/2-o(n)}, 2^{n/3-o(n)} \cdot \ell\right\}\right)$	[11, 19]

Table 1.2: Summary of the known results on linear secret-sharing schemes for sharing an ℓ-bit secret among n parties.

in undirected graphs (see Section 8.1 for background on matroids). Brickell and Davenport showed that (1) if an access structure is ideal, then it is a port of a matroid, and (2) if an access structure is a port of a linear matroid, then the access structure is ideal. The latter result extends to multilinear matroids. We prove these results in Chapter 8. As there are matroids whose port does not have an ideal secret-sharing scheme [162], there is a gap between the above necessary and sufficient conditions. The exact characterization of ideal access structures is still open.

Secret-Sharing Schemes with Computational Security. In all the results we mentioned so far, the security was information-theoretic, i.e., an unbounded adversary cannot learn any information on the secret. To decrease the share size, computational secret-sharing schemes were also considered [186, 117, 52, 32, 116, 7, 1, 20]. In computational secret-sharing schemes, the security only holds against a polynomial-time adversary (as common in cryptography). Furthermore, in computational secret-sharing schemes, we also require that the sharing and reconstruction algorithms run in polynomial time (as discussed in Section 4.5, linear secret-sharing schemes with polynomial-size shares also have this property). In Chapter 9, we describe four constructions of computational secret-sharing schemes:

1. A threshold t-out-of-n secret-sharing scheme of Krawczyk [117] with information ratio $O(1/t)$, i.e., the secret is an ℓ-bit string for a moderately large ℓ and each share is an $O(\ell/t)$-bit string. In every information-theoretic secret-sharing schemes the information ratio is at least 1.

2. A secret-sharing scheme of yao [186] for every access structure whose share size is the size of a *monotone circuit* representing the access structure, where the size of a circuit is the number of wires

in the circuit; this scheme assume the existence of one-way functions. Information-theoretic secret-sharing schemes are only known for *monotone formulas*.

3. A secret-sharing scheme of Komargodski, Naor, and Yogev [116] for every monotone access structure in which the share size is polynomial in the size of a *non-monotone* circuit representing the access structure;[3] this scheme assume the existence of witness-encryption schemes [91] and one-way functions. Compared to Yao's construction, the scheme of Komargodski et al. is much more efficient for some access structures since non-monotone circuits can be much smaller than monotone circuits [154]; however, the assumption used by Komargodski et al. is stronger.

4. A secret-sharing scheme of [7] for the Csirmaz access structure in which the share size is $O(\lambda)$, where λ is the security parameter. In every information-theoretic scheme, the share size for this access structure is $\Omega(n/\log(n)$ [66]; this is the biggest provable separation we can currently prove.

A recent result of Applebaum et al. [7] showed that every access structure can be realized by a computational secret-sharing scheme with polynomial share size (under the RSA assumption). The running time of the sharing and reconstruction in the scheme of [7] is exponential; the security of the scheme is against an exponential-time adversary. This result is not described in this monograph. Results of [121] imply that for almost all access structures, the reconstruction must require exponential time.

1.2 Comparison to an Earlier Version of This Monograph and Other Surveys

An earlier version of this monograph [15] was published in the IWCC conference in 2011. The current version reflects the advances in the area of secret sharing in the last decade and expands its contents. Specifically, we added a proof of a lower bound of $\log(n)$ on the share size of threshold secret-sharing schemes, a construction of secret-sharing schemes with share size 2^{cn} for $c < 1$, a chapter on computational secret-sharing schemes and a discussion on ideal secret-sharing schemes and their characterization via matroids.

There are other surveys on secret-sharing schemes, starting with the survey of Stinson [168], the book of Cramer, Damgård, and Nielsen [65], the lecture notes of Padró [145], the new book of Krenn and Thomas Lorünser [118], and the survey of Chattopadhyay, Saha, Nag, Nandi, [56]. Each survey has its own perspective and covers different subjects. This monograph focuses on secret-sharing for general access structures.

1.3 Organization

This monograph is intended for readers with some background in complexity, cryptography, and information theory; we provide the required definitions in Appendix A.1, Appendix A.2, and Appendix A.3 respectively.

[3] The result of Komargodski et al. [116] is stronger and applies to *non-deterministic* circuits representing the access structure, where a set of parties can efficiently reconstruct the secret if it has a witness that the circuits accepts the set.

The rest of the monograph is organized as follows. In Chapter 2 we discuss threshold secret-sharing schemes. In Chapter 3 we define secret-sharing schemes for general access structures, giving two definitions and proving that they are equivalent. In Chapter 4, we present efficient constructions of (linear) secret-sharing schemes and in Chapter 5 we present a fairly simple secret-sharing scheme for an arbitrary access structure with share size 2^{cn} for a constant $0 < c < 1$. In Chapter 6, we show how to construct secure multiparty protocols for general functions (in the semi-honest model) using secret-sharing schemes. In Chapter 7, we discuss lower bounds for secret-sharing schemes and present the best known lower bounds for general secret-sharing schemes and super-polynomial lower bounds for linear secret-sharing schemes. In Chapter 8, we discuss ideal secret-sharing schemes – the most efficient schemes in which the size of each share is the size of the secret. In Chapter 9, we discuss computational secret-sharing schemes and provide constructions of such schemes. Finally, in Chapter 10, we summarize this monograph and mention the most important open problems for secret sharing.

Chapters 5 to 9 are independent of each other. We recommend that a reader that is interested in one or more of these chapters will start by reading Chapters 2 to 4 (possibly excluding Sections 2.3, 2.4 and 4.6).

Chapter 2

Threshold Secret-Sharing Schemes

In this chapter, we discuss threshold secret-sharing schemes [40, 163] – the secret-sharing schemes that are mainly used.

2.1 The Definition of *t*-out-of-*n* Secret Sharing

We next define threshold secret-sharing schemes. In Chapter 3 we will generalize this definition to secret-sharing schemes realizing arbitrary access structures. We start by defining a secret-sharing scheme, which is a randomized mapping whose input is a string, called the secret, and output is n strings, called shares.

Definition 2.1 (Secret-Sharing Schemes)**.** *Let $\{p_1, \ldots, p_n\}$ be a set of parties. A* secret-sharing scheme $\Sigma = \langle \Pi, \mu \rangle$ *with domain of secrets S is a pair, where μ is a probability distribution over some finite set R, called the set of random strings, and Π is a mapping from $S \times R$ to a set of n-tuples $S_1 \times S_2 \times \cdots \times S_n$, where S_j is called the* domain of shares *of p_j. We denote the shares by $\langle \mathsf{sh}_1, \ldots, \mathsf{sh}_n \rangle$. For a set $A \subseteq \{p_1, \ldots, p_n\}$, we denote $\Pi_A(s; r)$ as the restriction of $\Pi(s; r)$ to its A-entries, i.e. $\langle \mathsf{sh}_i \rangle_{p_i \in A}$. The default distribution μ is the uniform distribution. In this case, we will simply denote the scheme by the mapping Π.*

Informally, we will consider a dealer that distributes a secret $s \in S$ according to Σ by first sampling a random string $r \in R$ according to μ, computing a vector of shares $\Pi(s; r) = \langle \mathsf{sh}_1, \ldots, \mathsf{sh}_n \rangle$, and privately communicating each share sh_j to party p_j.

In threshold secret-sharing schemes, every set whose size is at least some threshold t should be able to reconstruct the secret, while any smaller set should get no information on the secret.

Definition 2.2 (Threshold t-out-of-n Secret Sharing [163, 40])**.** *Let S be a finite set of secrets, where $|S| \geq 2$. A t-out-of-n secret-sharing scheme $\langle \Pi, \mu \rangle$ with a domain of secrets S is a secret-sharing scheme as defined in Definition 2.1 satisfying the following two requirements.*

Perfect Correctness. *The secret s can be reconstructed by any set of parties of size at least t. That is, for any set B such that $|B| \geq t$ (where $B = \{p_{i_1}, \ldots, p_{i_{|B|}}\}$), there exists a* reconstruction function

$\mathsf{Recon}_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ *such that for every* $s \in S$ *and every random string* r*,*

$$\mathsf{Recon}_B(\Pi_B(s; r)) = s. \tag{2.1}$$

Perfect Security. *Every set of size less than* t *cannot learn anything about the secret from its shares (in the information theoretic sense). Formally, for any set* T *such that* $|T| \leq t - 1$*, for every two secrets* $s_1, s_2 \in S$*, and for every possible vector of shares* $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$*:*

$$\Pr[\ \Pi_T(s_1; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}\ ] = \Pr[\ \Pi_T(s_2; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}\ ]. \tag{2.2}$$

2.2 Shamir's Threshold Secret-Sharing Scheme

Shamir [163] constructed a simple and elegant threshold scheme. In Shamir's scheme, the domain of secrets and shares is the elements of a finite field $\mathbb{F}_q$ for some prime-power $q > n$. The scheme is described in Figure 2.1. An illustration of Shamir's 2-out-of-n secret-sharing scheme is given in Figure 2.2.

Shamir's Secret-Sharing Scheme

The secret: an element $s \in \mathbb{F}_q$, where $q > n$ is a prime power.
The scheme:

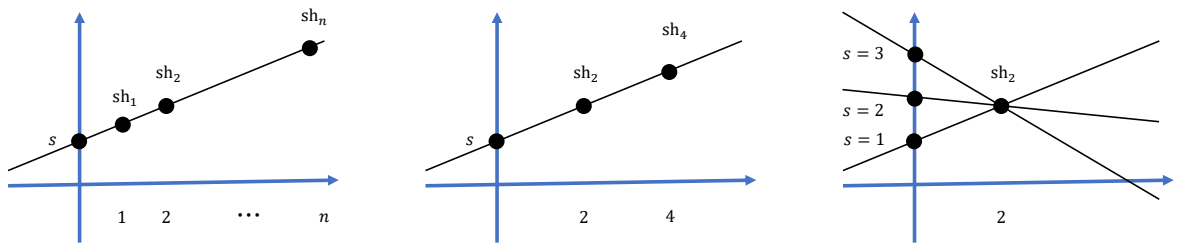
- Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be n distinct non-zero elements known to all parties (e.g., if $q > n$ is a prime, then we can take $\alpha_j = j$).
- Choose $t - 1$ random elements $a_1, \ldots, a_{t-1}$ from $\mathbb{F}_q$ independently with uniform distribution. These random elements together with the secret define a polynomial $P(x) \stackrel{\text{def}}{=} s + \sum_{i=1}^{t-1} a_i x^i$.
- The share of p_j is $\mathsf{sh}_j \leftarrow P(\alpha_j)$ (where P is evaluated using the arithmetic of $\mathbb{F}_q$).

Figure 2.1: Shamir's t-out-of-n secret-sharing scheme over a finite field $\mathbb{F}_q$, where $q > n$ is a prime-power.

The correctness and security of Shamir's scheme follow from the Lagrange's interpolation theorem:

Claim 2.3. *For every field* $\mathbb{F}$*, every* t *distinct values* $x_1, \ldots, x_t \in \mathbb{F}$*, and any* t *values* $y_1, \ldots, y_t$*, there exists a unique polynomial* Q *of degree at most* $t - 1$ *over* $\mathbb{F}$ *such that* $Q(x_j) = y_j$ *for* $1 \leq j \leq t$*. Furthermore, this polynomial can be efficiently computed.*

Lemma 2.4. *Let* t, n *be integers such that* $2 \leq t \leq$ *and let* q *be a prime-power such that* $q > n$*. Shamir's secret-sharing scheme, described in Figure 2.1, is a* t*-out-of-*n *secret-sharing scheme in which the secret and each share are elements in* $\mathbb{F}_q$*.*



(a) (b) (c)

Figure 2.2: An illustration of Shamir's 2-out-of-n secret-sharing scheme. Figure (a) demonstrates the sharing, where a random line $y = ax + s$ that passes through the secret is chosen and the share of p_i is the value of the line when $x = i$, i.e., $\mathsf{sh}_i = ai + s$. Figure (b) demonstrates the correctness, where given shares $\mathsf{sh}_2, \mathsf{sh}_4$ of p_2, p_4 respectively, the line through $(2, \mathsf{sh}_2)$ and $(4, \mathsf{sh}_4)$ is computed and the secret is the intersection of this line with the y-axis. Figure (c) demonstrates the security, where, for every share sh_2 of p_2, for every secret s there is a unique line that passes through $(0, s)$ and $(2, \mathsf{sh}_2)$. To simplify the figure, all lines are drawn over $\mathbb{R}$ rather than over a finite field.

Proof. To see that Shamir's scheme is correct, notice that every set B of size t holds t points of the polynomial P, hence we can reconstruct it using Lagrange's interpolation, and compute $s \leftarrow P(0)$. Formally, a set $B = \left\{ p_{i_1}, \ldots, p_{i_t} \right\}$ computes

$$Q(x) = \sum_{\ell=1}^{t} \mathsf{sh}_{i_\ell} \prod_{1 \le j \le t, j \ne \ell} \frac{\alpha_{i_j} - x}{\alpha_{i_j} - \alpha_{i_\ell}}.$$

Notice that $Q(\alpha_{i_\ell}) = \mathsf{sh}_{i_\ell} = P(\alpha_{i_\ell})$ for $1 \le \ell \le t$. That is, P and Q are polynomials of degree at most $t - 1$ that agree on t points, thus, by the uniqueness in the interpolation theorem (see Claim 2.3), P and Q are equal, and, in particular, $Q(0) = P(0) = s$. Thus, the parties in B reconstruct s by computing

$$s \leftarrow Q(0) = \sum_{\ell=1}^{t} \mathsf{sh}_{i_\ell} \prod_{1 \le j \le t, j \ne \ell} \frac{\alpha_{i_j}}{\alpha_{i_j} - \alpha_{i_\ell}}.$$

. For a given set B, the reconstruction function is a linear combination of the shares, that is,

$$s \leftarrow \sum_{\ell=1}^{t} \beta_\ell \cdot \mathsf{sh}_{i_\ell}, \quad \text{where } \beta_\ell = \prod_{1 \le j \le t, j \ne \ell} \frac{\alpha_{i_j}}{\alpha_{i_j} - \alpha_{i_\ell}}. \tag{2.3}$$

. Notice that $\beta_1, \ldots, \beta_t$ depend only on the set B and not on the secret s or the shares.

On the other hand, any unauthorized set T with $t - 1$ parties holds $t - 1$ points of the polynomial, which together with every possible secret (a value of the polynomial in the point 0) determines a unique polynomial of degree at most $t - 1$. Formally, by the interpolation theorem, for every $T = \left\{ p_{i_1}, \ldots, p_{i_{t-1}} \right\}$ and every $s \in \mathbb{F}_q$, there is a unique polynomial P_s with degree at most $t - 1$ such that $P_s(0) = s$ and $P_s(\alpha_{i_\ell}) = \mathsf{sh}_{i_\ell}$ for $1 \le \ell \le t - 1$. Hence, the probability that the shares $\left\langle \mathsf{sh}_{i_\ell} \right\rangle_{1 \le \ell \le t-1}$ are generated for the secret s is the

probability that P_s is selected, i.e.,

$$\Pr\left[\Pi_T(s;r) = \left\langle \mathsf{sh}_{i_\ell} \right\rangle_{1\le \ell \le t-1}\right] = \frac{1}{q^{t-1}}.$$

Since this probability is the same for every secret $s \in \mathbb{F}_q$, the security follows. □

2.3 Lower Bounds for Threshold Secret Sharing

In Shamir's scheme, each share (and the secret) is an element of a field with more than n elements, i.e., the share size is at least $\log(n)$. An interesting question is if the share size can be reduced when the secret is a bit. Kilian and Nisan [114] proved that this is not possible in every t-out-of-n secret-sharing scheme when t is not too close to n (e.g., $t \le n - \sqrt{n}$). Cascudo et al. [54] provided another proof of this result and generalized it to ramp secret-sharing schemes (i.e., schemes in which there is a gap between the reconstruction threshold and the security threshold). Bogdanov et al. [47] proved that the same bound holds for every $2 \le t \le n-1$. In this monograph, we present the lower bound proof of Kilian and Nisan. We first prove the result for 2-out-of-n secret-sharing schemes and then reduce the general case, i.e., a t-out-of-n scheme (when t is not too close to n) to the former case.

Lemma 2.5. *In any 2-out-of-n secret-sharing scheme, the size of the share of at least one party is at least* $\log(n)$ *even when the size of the domain of secrets is 2.*

Proof. W.l.o.g., $0, 1 \in S$. Assume that we share the secrets 0 and 1 independently; for every $1 \le i \le n$ denote these shares of p_i by $\Pi_i(0; r_0)$ and $\Pi_i(1; r_1)$ respectively. Consider the event EVENT_i, for $1 \le i \le n$, where EVENT_i occurs if and only if $\Pi_i(0; r_0) = \Pi_i(1; r_1)$, that is, in the two independent sharings of $s = 0$ and $s = 1$ respectively the share of p_i is the same. By the correctness of the 2-out-of-n secret-sharing scheme, these events are pairwise independent (otherwise, sometimes the shares of two parties are the same for the two secrets and it would be impossible to correctly reconstruct the secret for one of them). Thus, $\Pr[\vee_{1\le i\le n}\mathrm{EVENT}_i] = \sum \Pr_{1\le i\le n}[\mathrm{EVENT}_i]$. Since the probability of $\vee_{1\le i\le n}\mathrm{EVENT}_i$ is at most 1, there is at least one i such that

$$\Pr[\mathrm{EVENT}_i] = \Pr_{r_0,r_1}[\Pi_i(0; r_0) = \Pi_i(1; r_1)] \le 1/n.$$

Let $S_i = \{1, \ldots, \ell\}$ be the domain of shares of p_i and for $1 \le j \le \ell$ denote $\mathrm{pr}_j \stackrel{\text{def}}{=} \Pr_{r_0}[\Pi_i(0; r_0) = j]$. By the security requirement, the probability that $\mathsf{sh}_i = j$ is the same for both secrets, i.e., $\mathrm{pr}_j = \Pr_{r_1}[\Pi_i(1; r_1) =$

j]. Using this notation and the independence of the two sharings,

$$\begin{aligned}\Pr_{r_0,r_1}[\Pi_i(0;r_0)=\Pi_i(1;r_1)] &= \sum_{j=1}^{\ell}\Pr_{r_0,r_1}[\Pi_i(0;r_0)=j\wedge\Pi_i(1;r_1)=j]\\ &= \sum_{j=1}^{\ell}\left(\Pr_{r_0}[\Pi_i(0;r_0)=j]\cdot\Pr_{r_1}[\Pi_i(1;r_1)=j]\right)\\ &= \sum_{j=1}^{\ell}(\mathrm{pr}_j)^2.\end{aligned}$$

Thus, $\sum_{j=1}^{\ell}(\mathrm{pr}_j)^2 \leq 1/n$ and $\sum_{j=1}^{\ell}\mathrm{pr}_j = 1$. The minimum of the expression $\sum_{j=1}^{\ell}(\mathrm{pr}_j)^2$ is obtained when all probabilities are equal, i.e., $\sum_{j=1}^{\ell}(\mathrm{pr}_j)^2 \geq \sum_{j=1}^{\ell}(1/\ell)^2 = 1/\ell$. Thus, $1/n \geq \Pr[\mathrm{EVENT}_i] = \sum_{j=1}^{\ell}(\mathrm{pr}_j)^2 \geq 1/\ell$. We deduce that $\log(\ell)$ – the share size of p_i – is at least $\log(n)$ as claimed. □

Theorem 2.6. *In any t-out-of-n secret-sharing scheme, the size of the share of at least one party is at least* $\log(n-t+2)$.

Proof. We transform any t-out-of-n secret-sharing scheme Π^t to a 2-out-of-$(n-t+2)$ secret-sharing scheme Π^2 without increasing the share size; the theorem then follows from Lemma 2.5.

The transformation is simple – fix any possible vector of shares $\langle \mathsf{sh}_i\rangle_{n-t+3\leq i\leq n}$ of the last $t-2$ parties in Π^t; to share a secret $s\in\{0,1\}$ in Π^2 choose a random vector of shares of s in Π^t conditioned on the event that $\Pi^t_{\{p_{n-t+3},\ldots,p_n\}}(s;r) = \langle \mathsf{sh}_i\rangle_{n-t+3\leq i\leq n}$, that is,

$$\Pr[\Pi^2(s;r)=\langle \mathsf{sh}_i\rangle_{1\leq i\leq n-t+2}] = \frac{\Pr[\Pi^t(s;r)=\langle \mathsf{sh}_i\rangle_{1\leq i\leq n}]}{\Pr[\Pi^t_{\{p_{n-t+3},\ldots,p_n\}}(s;r)=\langle \mathsf{sh}_i\rangle_{n-t+3\leq i\leq n}]}.$$

The correctness of Π^2 follows from the correctness of Π^t, since any two parties in Π^2 hold two shares in Π^t and know the shares of $\{p_{n-t+3},\ldots,p_n\}$, i.e., they know t shares and can reconstruct the secret. The security of Π^2 follows from the security of Π^t, that is, for every secret s, $1\leq i\leq n-t+2$, and share sh_i of p_i:

$$\Pr[\Pi^2_{\{p_i\}}(s;r)=\langle \mathsf{sh}_i\rangle] = \frac{\Pr[\Pi^t_{\{p_i,p_{n-t+3},\ldots,p_n\}}(s;r)=\langle \mathsf{sh}_i,\mathsf{sh}_{n-t+3},\ldots,\mathsf{sh}_n\rangle]}{\Pr[\Pi^t_{\{p_{n-t+3},\ldots,p_n\}}(s;r)=\langle \mathsf{sh}_{n-t+3},\ldots,\mathsf{sh}_n\rangle]}; \tag{2.4}$$

since $|\{p_i,p_{n-t+3},\ldots,p_n\}|, |\{p_{n-t+3},\ldots,p_n\}| \leq t-1$, the probabilities in the denominator and numerator of (2.4) are independent of the secret and the security follows. □

2.4 Ramp Secret-Sharing Schemes

We will prove in Lemmas 7.1 and 7.2 that in any secret-sharing scheme the size of the share of each party is at least the size of the secret. This might be problematic if the size of the secret is large. Blakley and

Meadows [42] suggested a relaxation of threshold secret-sharing schemes that overcomes this problem. They defined (b, t)-ramp secret-sharing schemes, where $b < t$, in which every set of size at least t can reconstruct the secret, while every set of size at most b should not learn any information on the secret. For example, t-out-of-n threshold secret-sharing schemes are $(t-1, t)$-ramp secret-sharing schemes.

For long enough secrets, Blakley and Meadows constructed a (b, t)-ramp secret-sharing scheme with share size $1/(t-b)$ times the size of the secret. Chen and Cramer [58] and Chen et al. [59] showed that when the gap is large, namely $t - b = O(n)$, the share size can be reduced to $O(1)$ (that is, by bypassing the lower bound of [114, 47]). Cascudo, Cramer, and Xing [54] and Bogdanov, Guo, and Komargodski [47] proved lower bounds on the share size in ramp secret-sharing schemes – in any (b, t)-ramp secret-sharing scheme the size of at least one share is at least

$$\max\left\{\log\left(\frac{n-b+1}{t-b}\right), \log\left(\frac{t+1}{t-b}\right)\right\}.$$

Ramp schemes have found numerous applications in cryptography, e.g., [87, 169, 130]. Most notably, Franklin and Yung [87] and many follow-up works showed how to improve the communication complexity of secure multiparty computation (MPC) protocols using ramp secret-sharing schemes.

A (b, t)-Ramp Secret-Sharing Scheme

The secret: a vector $s = \langle s_1, \ldots, s_{t-b}\rangle \in \mathbb{F}_q^{t-b}$, where $q \geq n + t - b$ is a prime power.
The scheme:

- Let $\alpha_1, \ldots, \alpha_n, \alpha_{n+1}, \ldots, \alpha_{n+t-b} \in \mathbb{F}_q$ be $n + t - b$ distinct elements known to all parties (for example, if $q > n + t - b$ is a prime, then we can take $\alpha_j = j$).
- Choose a random polynomial $P(x)$ over $\mathbb{F}_q$ of degree at most $t-1$ such that $P(\alpha_{j+n}) = s_j$ for $1 \leq j \leq t - b$ (e.g. sample b uniformly distributed random elements $\mathsf{sh}_1, \ldots, \mathsf{sh}_b \in \mathbb{F}_q$ and find, using interpolation, the unique polynomial P such that $P(\alpha_j) = \mathsf{sh}_j$ for every $1 \leq j \leq b$ and $P(\alpha_{j+n}) = s_j$ for every $1 \leq j \leq t - b$).
- The share of p_j is $\mathsf{sh}_j \leftarrow P(\alpha_j)$ (where P is evaluated using the arithmetic of $\mathbb{F}_q$).

Figure 2.3: A (b, t)-ramp secret-sharing scheme over a finite field $\mathbb{F}_q$, where $q \geq n + t$ is a prime-power.

In Figure 2.3, we describe a (b, t)-ramp secret-sharing scheme in which the share size is $1/(t - b)$ times the size of the secret, i.e., when $t - b$ is big, the share size is much smaller than the size of the secret. This ramp scheme is a generalization of Shamir's scheme, where the secret is the evaluation of the polynomial in $t - b$ points. The correctness of the ramp scheme described in Figure 2.3 follows as in Shamir's secret-sharing scheme, i.e., every set of t parties can recover the polynomial and reconstruct the secret. Next, we argue that the scheme is secure. Consider a set B of b parties that hold shares $\langle \mathsf{sh}_i\rangle_{p_i \in B}$ and any secret $\langle s_1, \ldots, s_{t-b}\rangle \in \mathbb{F}_q^{t-b}$. By Claim 2.3, there is a unique polynomial P of degree at most $t - 1$ such that

$P(\alpha_j) = \mathsf{sh}_j$ for every $p_j \in B$ and $P(\alpha_{j+n}) = s_j$ for every $1 \leq j \leq t-b$, therefore, the probability of $\langle \mathsf{sh}_i \rangle_{p_i \in B}$ is the same for every possible secret.

We note that sets whose size is between $b+1$ and $t+1$ get partial information on the secret. For example, a set B of size $b+1$ knows $b+1$ values of the polynomial P; for every partial secret $s_1, \ldots, s_{t-b-1}$ it can compute the unique polynomial P such that $P(\alpha_j) = \mathsf{sh}_j$ for every $p_j \in B$ and $P(\alpha_{j+n}) = s_j$ for every $1 \leq j \leq t-b-1$ and conclude that $s_{t-b} = P(\alpha_{j+n})$; this implies that given the shares of B there are q^{t-b-1} possible vectors of secrets (out of the q^{t-b} vectors of secrets that were a priori possible).

Chapter 3

Definitions of Secret-Sharing Schemes for Arbitrary Access Structures

In this chapter we define secret-sharing schemes with information-theoretic security. We provide two definitions of the security of secret-sharing schemes and prove that they are equivalent.

Definition 3.1 (Access Structures)**.** *Let* $\{p_1, \ldots, p_n\}$ *be a set of parties. A collection* $\Gamma \subseteq 2^{\{p_1,\ldots,p_n\}}$ *is* monotone *if* $B \in \Gamma$ *and* $B \subseteq C$ *imply that* $C \in \Gamma$*. An* access structure *is a monotone collection* $\Gamma \subseteq 2^{\{p_1,\ldots,p_n\}}$ *of non-empty subsets of* $\{p_1, \ldots, p_n\}$*. Sets in* Γ *are called* authorized*, and sets not in* Γ *are called* unauthorized*.*

Example 3.2. Consider the access structure $\Gamma_{n,t} = \{B \subseteq \{p_1, \ldots, p_n\} : |B| \geq t\}$, i.e., all sets that contain at least t parties; this is the access structure of a t-out-of-n secret-sharing scheme. As another example, consider the access structure $\Gamma_{\sqcap}$ with 4 parties p_1, p_2, p_3, p_4

$$\Gamma_{\sqcap} \stackrel{\text{def}}{=} \{\{p_1, p_2\}, \{p_2, p_3\}, \{p_3, p_4\}\} \cup \{B \subseteq \{p_1, p_2, p_3, p_4\} : |B| \geq 3\}.$$

The sets $\{p_1, p_2\}, \{p_2, p_3\}, \{p_3, p_4\}$ of size two are authorized in $\Gamma_{\sqcap}$, while the sets $\{p_1, p_3\}, \{p_1, p_4\}$, $\{p_2, p_4\}$ of size two are unauthorized in $\Gamma_{\sqcap}$. We will study $\Gamma_{\sqcap}$ in Examples 4.2 and 4.16 and Theorem 7.4.

It is convenient to view an access structure as a function.

Definition 3.3. *We describe a set* $A \subseteq \{p_1, \ldots, p_n\}$ *by its characteristic vector (string)*

$$\mathbf{x_A} = \langle x_A[1], \ldots, x_A[n] \rangle \in \{0,1\}^n,$$

where $x_A[j] = 1$ *iff* $p_j \in A$*. Similarly, given an input* $x \in \{0,1\}^n$*, we denote* $I_x = \{p_i : x_i = 1\}$*.*

We represent an n-party access structure Γ *by the Boolean function* $f_\Gamma : \{0,1\}^n \to \{0,1\}$*, where* $f_\Gamma(\mathbf{x_B}) = 1$ *iff* $B \in \Gamma$*. We say that* f_Γ *represents* Γ*. As an access structure* Γ *is monotone, the function* f_Γ *is monotone.*[4]

[4] A function is monotone if for every $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \{0,1\}^n$ such that $x_i \leq y_i$ for every $1 \leq i \leq n$ it must be that $f(x_1, \ldots, x_n) \leq f(y_1, \ldots, y_n)$.

Recall that in Definition 2.1 we defined a secret-sharing scheme. We next define the correctness and perfect security of a secret-sharing scheme realizing a general access structure; we require that such scheme is secure against an unbounded adversary, i.e., its security is information-theoretic. The definition is based on [62, 18] and does not assume any probability distribution on the secrets.

Definition 3.4 (Secret-Sharing Schemes Realizing an Access Structure)**.** *Let S be a finite set of secrets, where $|S| \geq 2$. A secret-sharing scheme $\langle \Pi, \mu \rangle$ with domain of secrets S realizes an access structure Γ if the following two requirements hold:*

Perfect Correctness. *The secret s can be reconstructed by any authorized set of parties. That is, for any set $B \in \Gamma$ (where $B = \{p_{i_1}, \ldots, p_{i_{|B|}}\}$), there exists a* reconstruction function $\mathsf{Recon}_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ *such that for every $s \in S$ and random string $r \in R$,*

$$\mathsf{Recon}_B(\Pi_B(s; r)) = s. \tag{3.1}$$

Perfect Security. *Every unauthorized set cannot learn anything about the secret from its shares (in the information theoretic sense). Formally, for any set $T \notin \Gamma$, for every two secrets $s_1, s_2 \in S$, and for every possible vector of shares $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$:*

$$\Pr[\, \Pi_T(s_1; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] = \Pr[\, \Pi_T(s_2; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,]. \tag{3.2}$$

Remark 3.5. In the above definition, we required correctness with probability 1 and perfect security: for every two secrets s_1, s_2 the distributions $\Pi_T(s_1; r)$ and $\Pi_T(s_2; r)$ are identical. We can relax these requirements and require that the correctness holds with high probability and that the statistical distance between $\Pi_T(s_1; r)$ and $\Pi_T(s_2; r)$ is small.[5] To formalize this requirement, we say that a function $\mathrm{negl} : \mathbb{N} \to [0, 1]$ is negligible if for every $c \in \mathbb{N}$ there is an $n_c \in \mathbb{N}$ such that $\mathrm{negl}(\lambda) \leq 1/\lambda^c$ for every $\lambda > n_c$. In more details, we add a security parameter λ to Π and define the following requirements.

Statistical Correctness. The secret s can be reconstructed with probability almost 1 by any authorized set of parties. That is, there exists a negligible function $\mathrm{negl}(\lambda)$ such that for any set $B \in \Gamma$ (where $B = \{p_{i_1}, \ldots, p_{i_{|B|}}\}$), there exists a *reconstruction function* $\mathsf{Recon}_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ such that for every $s \in S$,

$$\Pr[\, \mathsf{Recon}_B(\Pi_B(1^\lambda, s; r)) = s \,] \geq 1 - \mathrm{negl}(\lambda). \tag{3.3}$$

Statistical Security. Every unauthorized set cannot learn non-negligible information about the secret from their shares. Formally, there is a negligible function $\mathrm{negl}(\lambda)$ such that for any set $T \notin \Gamma$ and for every two secrets $s_1, s_2 \in S$,

$$\mathrm{SD}\left(\Pi_T(1^\lambda, s_1; r), \Pi_T(1^\lambda, s_2; r)\right) \leq \mathrm{negl}(\lambda). \tag{3.4}$$

[5] The definition of statistical distance can be found in Definition A.2.

Schemes that satisfy these relaxed requirements are called statistical secret-sharing schemes. For example, such schemes are designed in [24]. A further relaxation of secret-sharing schemes with computational security is discussed in Chapter 9.

The most important complexity measure that we study in secret-sharing schemes in the share size.

Definition 3.6 (Share Size)**.** *The size of the secret in a secret-sharing scheme* $\Pi : S \times R \to S_1 \times \cdots \times S_n$ *is* $\log(|S|)$*, the share size of party* p_i *is* $\log(|S_i|)$*, the max share size is* $\max_{1 \le j \le n} \log(|S_j|)$*, and the total share size is* $\sum_{1 \le j \le n} \log(|S_j|)$*. The* information ratio *of a secret-sharing scheme is* $\frac{\max_{1 \le j \le n} \log(|S_j|)}{\log(|S|)}$ *and the* total information ratio *of a secret-sharing scheme is* $\frac{\sum_{1 \le j \le n} \log(|S_j|)}{\log(|S|)}$ *(informally, the information ratio measures the number of bits in the shares per a bit of the secret).*

We next define an alternative definition of secret-sharing schemes originating in [112, 53]; this definition uses the entropy function. For this definition we assume that there is some known probability distribution on the domain of secrets S. Any probability distribution on the secrets, together with the secret-sharing scheme $\langle \Pi, \mu \rangle$, induces, for any $A \subseteq \{p_1, \ldots, p_n\}$, a probability distribution on the vector of shares of the parties in A. We denote the random variable taking values according to this probability distribution on the vector of shares of A by $\mathcal{S}_A$, and denote the random variable denoting the secret by $\mathcal{S}$. The security in the alternative definition requires that if $T \notin \Gamma$, then the random variables $\mathcal{S}$ and $\mathcal{S}_T$ are independent. As traditional in the secret sharing literature, we formalize the above two requirements using the Shannon entropy function. For the definition of the entropy and some of its properties, see Appendix A.3.

Definition 3.7 (Secret-Sharing Schemes Realizing an Access Structure – Alternative Definition)**.** *We say that a secret-sharing scheme realizes an access structure* Γ *with respect to a given probability distribution on the secrets, denoted by a random variable* $\mathcal{S}$*, if the following conditions hold.*

PERFECT CORRECTNESS. *For every authorized set* $B \in \Gamma$*, the shares of the parties in* T *determine the secret, i.e.,*

$$H(\mathcal{S}|\mathcal{S}_B) = 0. \tag{3.5}$$

PERFECT SECURITY. *For every unauthorized set* $T \notin \Gamma$*, the shares of the parties in* T *and the secret are statistically independent, that is,*

$$H(\mathcal{S}|\mathcal{S}_T) = H(\mathcal{S}). \tag{3.6}$$

Definition 3.4 and Definition 3.7 are equivalent, as proved below in Claim 3.8. The advantage of Definition 3.4 is that it does not assume that there is a probability distribution on the secrets and that this distribution is known. Furthermore, Definition 3.4 can be generalized to statistical secret sharing[6] and computational secret sharing. On the other hand, Definition 3.7 is more convenient for proving lower bounds. Thus, the equivalence of the definitions allows choosing the more suitable definition for the specific task.

[6]One can suggest requiring that the entropy of the secret given the shares of the set T is high, i.e., $H(\mathcal{S}|\mathcal{S}_T) \approx H(\mathcal{S})$. However, such definition is distribution dependent; as argued above this is problematic. More importantly, although $H(\mathcal{S}|\mathcal{S}_T)$ might be large, it is possible that the scheme leaks the information that is important in the cryptographic application. For example, an attacker can know that the secret is either 0 or 1 and, seeing the shares of T, the attacker can distinguish between these secrets.

Furthermore, the equivalence of the definitions allows proving a result of Blundo et al. [45] that the security of a scheme according to Definition 3.7 is actually independent of the distribution: 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. The later property is important since when designing a secret-sharing scheme, usually one does not know the distribution of the secrets that will be used when the scheme is executed.

Claim 3.8. *The following claims are equivalent for a secret-sharing scheme $\langle \Pi, \mu \rangle$ realizing an access structure Γ:*

1. *The scheme $\langle \Pi, \mu \rangle$ is secure according to Definition 3.4.*
2. *There is some distribution on the secrets with support S (that is, $\Pr[S = s] > 0$ for every $s \in S$) such that the scheme is secure according to Definition 3.7.*
3. *For every distribution on the secrets whose support is contained in S, the scheme is secure according to Definition 3.7.*

Proof. We first show that Item 1 implies Item 3 (and, hence, Item 2). Let Π be a secret-sharing scheme that is secure according to Definition 3.4, and let S be a random variable distributed according to some distribution over S. Thus, for any set $T \notin \Gamma$, any secret $s_0 \in S$, and any shares $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$ for the parties in T,

$$\begin{aligned}
\Pr_{r,s}[\, S_T = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \mid S = s_0 \,] &= \Pr_{r}[\, \Pi_T(s_0; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] \\
&= \Pr_{r}[\, \Pi_T(s_0; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] \cdot \sum_{s \in S} \Pr_{s}[S = s] \\
&= \sum_{s \in S} \Pr_{s}[S = s] \cdot \Pr_{r}[\, \Pi_T(s_0; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] \\
&= \sum_{s \in S} \Pr_{s}[S = s] \cdot \Pr_{r}[\, \Pi_T(s; r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] \qquad (3.7) \\
&= \sum_{s \in S} \Pr_{s}[S = s] \cdot \Pr_{r,s}[\, S_T = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \mid S = s \,] \\
&= \Pr_{r,s}[\, S_T = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,], \qquad (3.8)
\end{aligned}$$

where the equality in (3.7) follows from (3.2). Thus, by (3.8), S_T and S are independent random variables, and, by the properties of the entropy function, $H(S|S_T) = H(S)$, thus, the scheme is secure according to Definition 3.7 with respect to this distribution on S.

Now assume that Π is a secret-sharing scheme which is secure according to Definition 3.7 for some fixed distribution on the secrets with support S, that is, assume that Item 2 holds and for any set $T \notin \Gamma$, $H(S|S_T) = H(S)$. This implies that for every secret s_0

$$\Pr_{r,s}[S = s_0 |\, \Pi_T(s; r) = S_T = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T} \,] = \Pr_{r,s}[S = s_0]. \qquad (3.9)$$

Hence,

$$\begin{aligned}
\Pr_r\left[\ \Pi_T(s_0;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ \right] &= \Pr_{r,s}[\ \Pi_T(s;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T} \mid \mathcal{S} = s_0\ ] \\
&= \frac{\Pr_{r,s}[\mathcal{S} = s_0 |\ \Pi_T(s;r) = \mathcal{S}_T = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ] \cdot \Pr_{r,s}[\Pi_T(s;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ]}{\Pr_{r,s}[\mathcal{S} = s_0]} \\
&= \Pr_{r,s}[\Pi_T(s;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ],
\end{aligned}$$

where the last equality follows from (3.9). This implies that for every two secrets s_1, s_2

$$\Pr_r[\Pi_T(s_1;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ] = \Pr_{r,s}[\Pi_T(s;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ] = \Pr_r[\Pi_T(s_2;r) = \left\langle \mathsf{sh}_j \right\rangle_{p_j\in T}\ ];$$

thus, the scheme is secure according to Definition 3.4. □

Chapter 4

Linear Secret-Sharing Schemes – Efficient Secret Sharing for Specific Access Structures

Linear secret-sharing schemes are secret-sharing schemes in which the secret is an element of a finite field and the shares are a linear combination of the secret and random elements from the field. They provide efficient secret-sharing schemes for access structures that have a small representation, e.g., if an access structure can be represented by a small monotone formula, then the access structure can be realized by an efficient linear secret-sharing scheme.[7] Furthermore, linear secret-sharing schemes are additive (also called homomorphic) – if the parties have shares of two secrets and each party sums these shares, then the parties hold shares for the sum of the secrets. In many applications of secret-sharing schemes, this additivity is essential (e.g., [64, 10, 182]).

Definition 4.1 (Linear Secret-Sharing Schemes)**.** *Let* $\Pi : S \times R \to S_1 \times \cdots \times S_n$ *be a secret-sharing scheme as defined in Definition 2.1. We say that* Π *is a* linear *secret-sharing scheme over a finite field* $\mathbb{F}_q$ *if there are integers* $\ell_r, \ell_1, \ldots, \ell_n$ *such that:*

- *The domain of secrets is* $S = \mathbb{F}_q$*,*
- *The randomness is chosen from* $R = \mathbb{F}_q^{\ell_r}$ *with uniform distribution,*[8]
- *The domains of shares are* $S_1 = \mathbb{F}_q^{\ell_1}, \ldots, S_n = \mathbb{F}_q^{\ell_n}$*, that is, the share of party* p_i *is composed of* ℓ_i *field elements,*
- *The mapping* Π *is a linear mapping over* $\mathbb{F}_q$ *from* $\mathbb{F}_q \times \mathbb{F}_q^{\ell_r}$ *to* $\mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$*, that is, its input is the secret and the* ℓ_r *elements in the random string and its output are the n shares.*

[7] The best known secret-sharing schemes for arbitrary access structures are non-linear, however they are not efficient since their share size is exponential.

[8] Without this requirement we would be able to use a non-linear mapping to share a secret by using "strange distributions".

We start with an example of a linear secret-sharing scheme.

Example 4.2. We describe a linear scheme realizing $\Gamma_{\sqcap}$ (defined in Example 3.2) whose information ratio is 2, i.e., each party gets at most 2 field elements. Let q be any prime power. To share a secret $s \in \mathbb{F}_q$, the dealer independently chooses two random elements $r_1, r_2 \in \mathbb{F}_q$ with a uniform distribution. The share of p_1 is r_1, the share of p_2 is $r_1 + s$, the share of p_3 is two elements, r_1 and $r_2 + s$, and the share of p_4 is r_2; in this case $\ell_r = 2$, $\ell_1 = \ell_2 = \ell_4 = 1$, and $\ell_3 = 2$. It can be verified that this scheme realizes $\Gamma_{\sqcap}$. For example, p_2 and p_3 can reconstruct the secret since they hold $r_1 + s$ and r_1 respectively. On the other hand, p_1 and p_3 do not get any information on the secret since together they hold r_1 and $r_2 + s$, which are uniformly distributed regardless of the secret.

In this chapter we describe linear secret-sharing schemes and their properties. In Sections 4.1 to 4.3 we describe four specific constructions of linear secret-sharing schemes. These constructions preceded the general notion of linear secret-sharing schemes and are important for their own sake. In Section 4.4 we define monotone span programs and show how to construct linear secret-sharing from them; every linear secret-sharing scheme can be cast in this way. In Section 4.5 we discuss some useful properties of linear secret-sharing schemes. Finally, in Section 4.6 we present a generalization of linear secret-sharing schemes, called multilinear secret-sharing schemes; in these schemes the secret can be composed of more than one field element.

4.1 Undirected *s*-*t*-Connectivity

In this section we describe a simple and elegant construction of a linear secret-sharing scheme of Benaloh and Rudich [35]. Consider the access structure Γ_{ustcon}, whose parties correspond to *edges* of a complete undirected graph with m vertices $v_1, \ldots, v_m$, that is, there are $n = \binom{m}{2}$ parties in the access structure and a party is an edge (v_i, v_j), where $i < j$. A set of parties (edges) is in the access structure if the set contains a path from v_1 to v_m. Benaloh and Rudich [35] constructed a secret-sharing scheme realizing this access structure. In Figure 4.1, we describe this secret-sharing scheme. In this scheme, there is a random bit r_i for each vertex v_i (where $r_1 = s$ is the secret and $r_m = 0$); the share of the edge (v_i, v_j) is $r_i \oplus r_j$.

Lemma 4.3. *The scheme described in Figure 4.1 is a linear secret-sharing scheme realizing the access structure* Γ_{ustcon}*, where the secret and each share are bits.*

Proof. To see that the scheme described in Figure 4.1 is correct, consider a set of parties that form a path $v_1 = v_{i_1}, v_{i_2}, \ldots, v_{i_{\ell-1}}, v_{i_\ell} = v_m$, and consider the exclusive or of the shares given to the parties (edges) of the path:

$$(r_{i_1} \oplus r_{i_2}) \oplus (r_{i_2} \oplus r_{i_3}) \oplus \cdots \oplus (r_{i_{\ell-2}} \oplus r_{i_{\ell-1}}) \oplus (r_{i_{\ell-1}} \oplus r_{i_\ell}) = r_{i_1} \oplus r_{i_\ell} = r_1 \oplus r_m = s.$$

To see that the scheme described in Figure 4.1 is secure consider an unauthorized set, that is, a set of edges T not containing a path from v_1 to v_m. Define the set of vertices V_1 such that $v_i \in V_1$ if there exists

The ustcon secret-sharing scheme

The secret: a bit $s \in \{0,1\}$.

The scheme:

- Choose $m-2$ random bits $r_2, \ldots, r_{m-1}$ independently with uniform distribution.
- Set $r_1 \leftarrow s$ and $r_m \leftarrow 0$.
- The share of a party (v_i, v_j) is $r_i \oplus r_j$.

Figure 4.1: A secret-sharing scheme realizing the access structure Γ_{ustcon}.

a path in the graph (V,T) from v_1 to v_i. By definition, $v_1 \in V_1$ and $v_m \notin V_1$. Furthermore, for every $(v_i, v_j) \in T$ either both vertices v_i, v_j are in V_1 or both of them are not in V_1.

Let $\left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i,v_j)\in T}$ be shares generated for the parties in T with the secret $s=0$, where $\mathsf{sh}_{i,j}$ is the share given to the party (v_i, v_j). We next show that the number of vectors of random bits $r_2, r_3, \ldots, r_{m-1}$ that generate $\left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i,v_j)\in T}$ given the secret $s=0$ is equal to the number of vectors of random bits that generate these shares given the secret $s=1$. Fix a vector of random bits $r_2, r_3, \ldots, r_{m-1}$ that generates the shares $\left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i,v_j)\in T}$ with the secret $s=0$. Recall that $r_1 = s = 0$ and $r_m = 0$. Consider the random bits $r'_1, \ldots, r'_m$, where $r'_i = \overline{r_i}$ if $v_i \in V_1$ and $r'_i = r_i$ otherwise. First note that $r'_1 = \overline{r_1} = 1$ and $r'_m = r_m = 0$ as required for sharing $s=1$. We claim that the random bits $r'_2, \ldots, r'_{m-1}$ generate the shares $\left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i,v_j)\in T}$ with the secret $s=1$, that is, $\mathsf{sh}_{i,j} = r'_i \oplus r'_j$ for $(v_i, v_j) \in T$. There are only two cases to consider.

- For every $(v_i, v_j) \in T$ such that $v_i, v_j \in V_1$

$$r'_i \oplus r'_j = \overline{r_i} \oplus \overline{r_j} = r_i \oplus r_j = \mathsf{sh}_{i,j}.$$

- For every $(v_i, v_j) \in T$ such that $v_i, v_j \notin V_1$

$$r'_i \oplus r'_j = r_i \oplus r_j = \mathsf{sh}_{i,j}.$$

Notice that the mapping from $r_1, \ldots, r_m$ to $r'_1, \ldots, r'_m$ is invertible. To conclude, the number of vectors of random bits that generate the shares $\left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i,v_j)\in T}$ given the secret 0 is the same as the number of vectors of random bits that generate these shares given the secret 1. This implies that

$$\Pr[\Pi_T(0;r) = \left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i.v_j)\in T}] = \Pr[\Pi_T(1;r) = \left\langle \mathsf{sh}_{i,j} \right\rangle_{(v_i.v_j)\in T}];$$

thus, the scheme is secure.

This scheme is linear over the field with two elements $\mathbb{F}_2$.[9] In particular, the randomness is a vector $\left\langle r_2, \ldots, r_{|V|-1} \right\rangle$ of $|V|-2$ random elements in $\mathbb{F}_2$, the share of an edge (v_1, v_j) is $s \oplus r_j$ (a linear combination

[9] We can generalize this scheme to be linear over any finite field $\mathbb{F}_q$ (and, in fact any finite group). To share a secret $s \in \mathbb{F}_q$, the dealer chooses $m-2$ random elements $r_2, \ldots, r_{m-1}$ from $\mathbb{F}_q$ independently with uniform distribution, and sets $r_1 \leftarrow s$ and $r_m \leftarrow 0$. The share of a party (v_i, v_j), where $i < j$, is $r_j - r_i$.

where the coefficient of s and r_j are 1 and all other coefficients are zero), and the share of an edge (v_i, v_j) for $2 \leq i < j \leq m$ is $r_i \oplus r_j$ (where $r_m = 0$). □

Notice that the reconstruction of the secret in the scheme described in Figure 4.1 is a sum of the shares of the authorized set, i.e., it is a linear function. For example, the authorized set $\{(v_1, v_2), (v_2, v_3), (v_3, v_m)\}$, holding shares $\mathsf{sh}_{1,2} = s \oplus r_2$, $\mathsf{sh}_{2,3} = r_2 \oplus r_3$, $\mathsf{sh}_{3,m} = r_3$, reconstructs the secret by computing $\mathsf{sh}_{1,2} \oplus \mathsf{sh}_{2,3} \oplus \mathsf{sh}_{3,m}$, which results in s.

4.2 Ito, Saito, and Nishizeki's Constructions

Ito, Saito, and Nishizeki [107] defined secret-sharing schemes for general access structures and provided two constructions of schemes for an arbitrary access structure Γ. These schemes are linear.

First Construction. The first construction of Ito et al. shares the secret independently for each minimal authorized set using the scheme of Example 1.1. The scheme is described in Figure 4.2; it is known as the DNF secret-sharing scheme. For concreteness, the scheme can be executed over $\mathbb{F}_2$; in this case, the secret is a bit. We emphasize that for each minimal authorized set $B \in \Gamma$ the random elements are chosen by the dealer independently. Clearly, each set in $A \in \Gamma$ contains a minimal authorized set $B \subseteq A$ and can reconstruct the secret by computing the sum of the elements given to the set B. On the other hand, each unauthorized set $T \notin \Gamma$ misses at least one party from each minimal authorized set; thus, it misses at least one element given to the minimal authorized set. In other words, the elements held by the parties in T are uniformly distributed and independent of the secret.

To summarize, the number of elements that p_i gets is the number of minimal authorized sets that contain p_i. When the number of minimal authorized sets is small, this scheme is efficient, specifically when the field in $\mathbb{F}_2$. However, this scheme is highly inefficient for access structures in which the number of minimal sets is big.

We note that the fact that an access structure has many minimal authorized sets does not mean that it does not have an efficient secret-sharing scheme. For example, consider the $n/2$-out-of-n access structure, that is, the access structure

$$\Gamma_{n/2} \stackrel{\text{def}}{=} \left\{ B \subseteq \{p_1, \ldots, p_n\} \;:\; |B| \geq n/2 \right\}.$$

The number of bits that each party gets in the first ISN scheme, described in Figure 4.2, for the access structure $\Gamma_{n/2}$ is $\binom{n-1}{n/2-1} = \Theta(2^n/\sqrt{n})$. On the other hand, in Shamir's secret-sharing scheme for this access structure the size of each share is the same as the size of the secret.

Second Construction. The second construction of Ito et al. is dual to the first construction; in this scheme, the dealer ensures that every unauthorized set cannot reconstruct the secret. The scheme is described in Figure 4.3; it is known as the CNF secret-sharing scheme.

The first ISN secret-sharing scheme

The secret: an element $s \in \mathbb{F}_q$, for some finite field $\mathbb{F}_q$.
The scheme:

- For every minimal authorized set $B \in \Gamma$, where $B = \left\{ p_{i_1}, \ldots, p_{i_\ell} \right\}$ do:
  - Choose $\ell - 1$ random field elements $r_1^B, \ldots, r_{\ell-1}^B \in \mathbb{F}_q$ with uniform distribution.
  - Compute $r_\ell^B \leftarrow s - (r_1^B + \cdots + r_{\ell-1}^B)$ (where the sum is in $\mathbb{F}_q$).
  - Give p_{i_j} the element r_j^B.

Figure 4.2: The first ISN secret-sharing scheme. The share of each party p_i is an element for every authorized set containing p_i.

The second ISN secret-sharing scheme

The secret: an element $s \in \mathbb{F}_q$, for some finite field $\mathbb{F}_q$.
The scheme:

- Let $T_1, \ldots, T_\ell$ be the maximal unauthorized sets of the access structure Γ.
- Share s with the ℓ-out-of-ℓ secret-sharing scheme of Example 1.1, that is, choose $\ell - 1$ random field elements $r_1, \ldots, r_{\ell-1} \in \mathbb{F}_q$, and compute $r_\ell \leftarrow s - (r_1 + \cdots + r_{\ell-1})$.
- The share of party p_i are all the elements r_j such that $p_i \notin T_j$.

Figure 4.3: The second ISN secret-sharing scheme. The share of each party p_i is an element for every maximal unauthorized set not containing p_i.

We next discuss the correctness and security of the scheme. Let $A \in \Gamma$ be an authorized set. By the monotonicity of Γ, for every maximal unauthorized set T_j, it must be that $A \not\subseteq T_j$, i.e., there exists some $p_m \in A \setminus T_j$. Thus, the parties in A hold all elements r_j and can reconstruct the secret. On the other hand, each unauthorized set $T \notin \Gamma$ is a subset of at least one maximal unauthorized set T_j, hence they do not hold r_j. By the security of the ℓ-out-of-ℓ secret-sharing scheme, the parties in T get no information on the secret.

The number of elements that p_i gets is the number of maximal unauthorized sets that do not contain p_i. If the number of unauthorized sets is small, this scheme is efficient. Furthermore, for some access structures it can be exponentially more efficient than the first construction (and vice versa). Still, for the worst access structures, the size of the shares in this scheme is $\Theta(2^n/\sqrt{n})$.

As evident from the description of the first and second ISN schemes, both schemes are linear over $\mathbb{F}_q$. Furthermore, in both schemes the reconstruction of the secret is a linear combination of the shares.

4.3 The Monotone Formulas Construction

Benaloh and Leichter [36] describe a construction of secret-sharing schemes for any access structure based on monotone formulas. The construction of [36] generalizes the constructions of [107] and is more efficient for many access structures. However, also in this scheme for almost all access n-party structures the size of the shares is exponential in the number of parties even for a one-bit secret, i.e., the share size is $2^{n-o(n)}$.

The scheme of Benaloh and Leichter is recursive. It starts with schemes for simple access structures and constructs a scheme for a composition of the access structures; that is, it uses the following closure properties of secret-sharing schemes.

Lemma 4.4. *Let Γ_1, Γ_2 be access structures be two access structures over the same set of parties $\{p_1, \ldots, p_n\}$.*[10] *Assume that for $b \in \{1,2\}$ there is a secret-sharing scheme Π_b realizing Γ_b, where the two schemes have the same domain of secrets S and for every $1 \leq j \leq n$ the share of p_j in the scheme Π_b is an element in $S_{j,b}$. Then there exist secret-sharing schemes realizing $\Gamma_1 \cup \Gamma_2$ and $\Gamma_1 \cap \Gamma_2$ in which the domain of shares of p_j is $S_{j,1} \times S_{j,2}$. Furthermore, if Π_1 and Π_2 are linear over $\mathbb{F}_q$ for some finite field $\mathbb{F}_q$, then the resulting schemes are linear over $\mathbb{F}_q$.*

Proof. For the finite set S, let $\langle S, + \rangle$ be a group (e.g., assume without loss of generality that $S = \{0, \ldots, m-1\}$ for some $m \in \mathbb{N}$ and take $+$ as the sum modulo m).

To share a secret $s \in S$ for the access structure $\Gamma_1 \cup \Gamma_2$, independently share s using the scheme Π_i (realizing Γ_i) for $i \in \{1,2\}$; the share of p_i is its share in Π_1 and its share in Π_2. Clearly, if both Π_1 and Π_2 are linear over the same field, then the resulting scheme is linear.

To share a secret $s \in S$ for the access structure $\Gamma_1 \cap \Gamma_2$, choose $r_1 \in S$ with a uniform distribution and let $r_2 \leftarrow s - r_1$. Next, for $i \in \{1,2\}$, independently share r_i using the scheme Π_i (realizing Γ_i). For every set

[10] This assumption is technical (otherwise $\Gamma_1 \cup \Gamma_2$ or $\Gamma_1 \cap \Gamma_2$ might not be monotone). It is without loss of generality as it is possible that some parties are redundant in one of the access structures, that is, there might be parties that do not belong to minimal authorized sets in one of the access structures.

$B \in \Gamma_1 \cap \Gamma_2$, the parties in B can reconstruct both r_1 and r_2 and compute $s \leftarrow r_1 + r_2$. On the other hand, for every set $T \notin \Gamma$, the parties in T do not have any information on at least one r_i, hence do not have any information on the secret s. Notice that if both Π_1 and Π_2 are linear over $\mathbb{F}_q$ and we use the additive group of $\mathbb{F}_q$ as the group $(S, +)$, then the resulting scheme is linear; for example, in Π_2, we replace each linear combination of the secret by a linear combination of $s - r_1$. □

Example 4.5. Given an access structure Γ, whose minimal authorized sets are $\{B_1, \ldots, B_\ell\}$, we define $\Gamma_i \stackrel{\text{def}}{=} \{A : B_i \subseteq A\}$. Clearly, $\Gamma = \bigcup_{1 \leq i \leq \ell} \Gamma_i$, and for every $1 \leq i \leq \ell$ there is a scheme realizing Γ_i with a domain of secrets $\mathbb{F}_q$, where the share of each $p_j \in B$ is one field element. Thus, the closure properties of Lemma 4.4 imply the first scheme of Ito, Saito, and Nishizeki.

The second scheme of Ito, Saito, and Nishizeki is also implied by the closure properties of Lemma 4.4. Let $T_1, \ldots, T_\ell$ be the maximal unauthorized sets of an access structure Γ and define $\Gamma_i \stackrel{\text{def}}{=} \{B : B \nsubseteq T_i\}$ (that is, T_i is the only maximal unauthorized set in Γ_i). Then, $\Gamma = \bigcap_{1 \leq i \leq \ell} \Gamma_i$.

Benaloh and Leichter applied the closure properties recursively, providing efficient secret-sharing schemes for a much richer family of access structures than the access structures that can be efficiently realized by the scheme of Ito, Saito, and Nishizeki. To describe access structures that can be efficiently realized by Benaloh and Leichter's scheme, it is convenient to view an access structure Γ as a function f_Γ as defined in Definition 3.3. This definition implies that for two access structures Γ_1 and Γ_2, $f_{\Gamma_1} \vee f_{\Gamma_2} = f_{\Gamma_1 \cup \Gamma_2}$ and $f_{\Gamma_1} \wedge f_{\Gamma_2} = f_{\Gamma_1 \cap \Gamma_2}$. Using this observation, the scheme of Benaloh and Leichter (denoted the BL scheme) can efficiently realize every access structure that can be represented by a small monotone formula (the reader is referred to Appendix A.1 for a reminder on monotone formulas). This is achieved by recursively applying Lemma 4.4 to each internal node of the monotone formula. Notice that if we follow the recursion, each leaf gets one element; we give the element of a leaf labeled by x_i to party p_i. A formal description of the BL scheme is presented in Figure 4.4. An example of an execution of the BL scheme for a monotone formula is given in Figure 4.5.

Lemma 4.6. *Let Γ be an access structure and assume that can be represented by a monotone formula in which for every $1 \leq j \leq n$ the variable x_j appears a_j times in the formula. Then, for every prime-power q, the BL scheme described in Figure 4.4 is a linear secret-sharing scheme over $\mathbb{F}_q$ realizing the access structure Γ. The secret in the BL secret-sharing scheme is an element in $\mathbb{F}_q$ and for every $i \in \{1, \ldots, n\}$ the share of p_j is a_j elements in $\mathbb{F}_q$.*

Any monotone Boolean function over n variables can be computed by a monotone formula. Thus, every access structure can be realized by the scheme of [36]. However, for almost all monotone functions, the size of the smallest monotone formula computing them is exponential in n; i.e., the information ratio of the resulting scheme is exponential in the number of the parties. Note that we can consider additional types of gates in the monotone formula. If the gate can be realized by an efficient secret-sharing scheme (e.g., a threshold gate), then the resulting scheme will be efficient.

The BL secret-sharing scheme

Procedure ShareBL(s, F)

The inputs: a secret $s \in \mathbb{F}_q$, for some finite field $\mathbb{F}_q$, and a monotone formula F.

- If $F = x_i$ for some $1 \leq i \leq n$, then give s to p_i.
- If $F = \vee_{j=1}^{\ell} F_j$ for some monotone formulas $F_1, \ldots, F_\ell$, then ShareBL(s, F_j) for every $1 \leq j \leq \ell$.
- If $F = \wedge_{j=1}^{\ell} F_j$ for some monotone formulas $F_1, \ldots, F_\ell$, then
  - Choose $\ell - 1$ random field elements $r_1, \ldots, r_{\ell-1} \in \mathbb{F}_q$ with uniform distribution and compute $r_\ell \leftarrow s - (r_1 + \cdots + r_{\ell-1})$ (where the sum is in $\mathbb{F}_q$).
  - ShareBL(r_j, F_j) for every $1 \leq j \leq \ell$.

Figure 4.4: The BL secret-sharing scheme for an access structure represented by a monotone formula F. The share of each party p_i is an element for every leaf labeled by x_i.

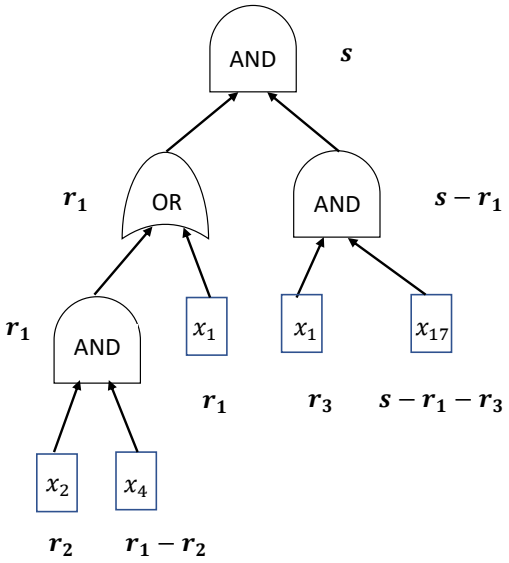


Figure 4.5: An example of an execution of the BL secret-sharing scheme for a monotone formula. The values given in the recursion to each node are in **bold**. For each AND gate, the scheme chooses a fresh random element. The share of p_1, for example, is the two elements given to the leaves labeled by x_1, i.e., $\mathsf{sh}_1 = \langle r_1, r_3 \rangle$.

4.4 Linear Secret-Sharing Schemes via Monotone Span Programs

In this section, we discuss linear secret-sharing schemes in their generality. To model a linear scheme, we use *monotone span programs* [111], which is, basically, the matrix describing the linear mapping of the linear scheme. The monotone span program also defines the access structure that the secret-sharing scheme realizes. In the rest of the paper, vectors are denoted by bold letters (e.g., $\mathbf{r}$) and, according to the context, vectors are either row vectors or column vectors (i.e., if we write $\mathbf{r}\,\boldsymbol{M}$, then $\mathbf{r}$ is a row vector, if we write $\boldsymbol{M}\mathbf{r}$, then $\mathbf{r}$ is a column vector).

Definition 4.7 (Monotone Span Programs [111])**.** *A monotone span program is a triple* $\mathsf{MSP} = \langle \mathbb{F}, \boldsymbol{M}, \rho \rangle$*, where* $\mathbb{F}$ *is a field,* $\boldsymbol{M}$ *is an* $a \times b$ *matrix over* $\mathbb{F}$ *for some* $a, b \in \mathbb{N}$*, and* $\rho : \{1, \ldots, a\} \to \{p_1, \ldots, p_n\}$ *labels each row of* $\boldsymbol{M}$ *by a party.*[11] *The size of* MSP *is the number of rows of* $\boldsymbol{M}$ *(i.e.,* a*). For any set* $A \subseteq \{p_1, \ldots, p_n\}$*, let* $\boldsymbol{M}_A$ *denote the sub-matrix obtained by restricting* $\boldsymbol{M}$ *to the rows labeled by parties in* A*. We say that* MSP *accepts* B *if the rows of* $\boldsymbol{M}_B$ *span the vector* $\mathbf{e_1} \stackrel{def}{=} \langle 1, 0, \ldots, 0 \rangle$*. We say that* MSP *accepts an access structure* Γ *if* MSP *accepts a set* B *iff* $B \in \Gamma$*.*

Example 4.8. Consider the following monotone span program $\langle \mathbb{F}_{17}, \boldsymbol{M}, \rho \rangle$, where

$$\boldsymbol{M} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix} \quad \begin{aligned} \rho(1) &= p_2 \\ \rho(2) &= p_2 \\ \rho(3) &= p_1 \\ \rho(4) &= p_3 \end{aligned}$$

Consider the sets $B = \{p_1, p_2\}$ and $T = \{p_1, p_3\}$. In this case

$$\boldsymbol{M}_B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \quad \text{and} \quad \boldsymbol{M}_T = \begin{pmatrix} 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}.$$

As $\boldsymbol{M}_B$ has full rank, the rows of $\boldsymbol{M}_B$ span $\mathbf{e_1}$, i.e., $\langle 3, 14, 1 \rangle\, \boldsymbol{M}_B = \mathbf{e_1}$ (in $\mathbb{F}_{17}$). Hence, the monotone span program accepts $\{p_1, p_2\}$. On the other hand, the rows of $\boldsymbol{M}_T$ do not span $\mathbf{e_1}$ and the monotone span program does not accept $\{p_1, p_3\}$. The minimal authorized sets in the access structure accepted by MSP are $\{p_1, p_2\}$ and $\{p_2, p_3\}$.

A monotone span program implies a linear secret-sharing scheme for an access structure containing all the sets accepted by the program as stated below.

Claim 4.9 ([49, 111])**.** *Let* $\mathsf{MSP} = \langle \mathbb{F}_q, \boldsymbol{M}, \rho \rangle$ *be a monotone span program accepting an access structure* Γ*, where* $\mathbb{F}_q$ *is a finite field. Assume that for every* $j \in \{1, \ldots, n\}$ *there are* a_j *rows of* $\boldsymbol{M}$ *labeled by* p_j*. Then, there is a linear secret-sharing scheme realizing* Γ *such that the share of party* p_j *contains* a_j *elements of* $\mathbb{F}_q$*. The information ratio of the resulting scheme is* $\max_{1 \le j \le n} a_j$*.*

The MSP secret-sharing scheme

The secrets: an element $s \in \mathbb{F}_q$, for some prime power q.
The monotone span program: $\mathsf{MSP} = \left\langle \mathbb{F}_q, M, \rho \right\rangle$, where M is an $a \times b$ matrix over $\mathbb{F}_q$.
The scheme:

- Choose $b-1$ random elements $r_2, \ldots, r_b$ independently with uniform distribution from $\mathbb{F}_q$ and let $\mathbf{r} \leftarrow \langle s, r_2, \ldots, r_b \rangle$.
- Evaluate $\langle z_1, \ldots, z_a \rangle \leftarrow M\mathbf{r}$; the share of party p_j are the a_j entries corresponding to rows labeled by p_j, i.e., $\mathsf{sh}_j = \langle z_i \,:\, \rho(i) = j \rangle$.

Figure 4.6: The linear secret-sharing scheme for a monotone span program MSP.

Proof. The linear secret-sharing scheme for Γ is described in Figure 4.6. In this secret-sharing scheme, every set $B \in \Gamma$ can reconstruct the secret: Let $N = M_B$, thus, the rows of N span $\mathbf{e_1}$, and there exists some vector $\mathbf{v}$ such that $\mathbf{e_1} = \mathbf{v}N$. Notice that the first coordinate in r is s and the shares of the parties in B are $N\mathbf{r}$. The parties in B can reconstruct the secret by computing $\mathbf{v}(N\mathbf{r})$ since

$$\mathbf{v}(N\mathbf{r}) = (\mathbf{v}N)\mathbf{r} = \mathbf{e_1} \cdot \mathbf{r} = s. \tag{4.1}$$

We next prove that this scheme is secure. If $T \notin \Gamma$, then the rows of M_T do not span the vector $\mathbf{e_1}$, i.e., $\operatorname{rank}(M_T) < \operatorname{rank}\binom{M_T}{\mathbf{e_1}}$ (where $\binom{M_T}{\mathbf{e_1}}$ is the matrix containing the rows of M_T and an additional row $\mathbf{e_1}$). By simple linear algebra, $\left|\operatorname{kernel}(M_T)\right| > \left|\operatorname{kernel}\binom{M_T}{\mathbf{e_1}}\right|$, and there is some vector $\mathbf{w} \in \mathbb{F}_q^b$ such that $(M_T)\mathbf{w} = \mathbf{0}$ and $\mathbf{e_1} \cdot \mathbf{w} = 1$ (that is, w_1 – the first coordinate in $\mathbf{w}$ – is 1). We next prove that for every two secrets $s_0, s_1 \in \mathbb{F}_q$ and every vector of shares $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$ for the parties in T, the probability that the scheme outputs this vector of shares is the same for the two secrets. Fix a vector $\mathbf{r} \stackrel{\text{def}}{=} \langle s_0, r_2, \ldots, r_b \rangle$ such that $(M_T)\mathbf{r} = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$, that is, $\mathbf{r}$ is a vector generating the shares for the secret s_0. Consider the vector $\mathbf{r'} = \mathbf{r} + (s_1 - s_0)\mathbf{w}$. Let r_1 and r'_1 be the first coordinate in $\mathbf{r}$ and $\mathbf{r'}$ respectively. As $r_1 = s_0$ and $w_1 = 1$,

$$r'_1 = r_1 + (s_1 - s_0)w_1 = s_1;$$

thus, the vector $\mathbf{r'}$ generates shares for the secret s_1. Furthermore, as $\mathbf{w}$ is in the kernel of M_T,

$$(M_T)\mathbf{r'} = (M_T)(\mathbf{r} + (s_1 - s_0)\mathbf{w}) = (M_T)\mathbf{r} + (s_1 - s_0)(M_T)\mathbf{w} = (M_T)\mathbf{r} = \left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}.$$

The mapping $\mathbf{r'} = \mathbf{r} + (s_1 - s_0)\mathbf{w}$ from random strings for the secret s_0 to random strings for the secret s_1 is invertible, that is, the number of random strings that generate the shares $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$ when the secret is s_0

[11] For simplicity, in this monograph we label a row by a party p_j rather than by a variable x_j as done in [111]. In [111], they also define general (non-monotone) span programs, where a row can be also labeled by a negated variable $\overline{x_j}$ and M_B is defined appropriately.

is the same as the number of random strings that generate these shares when the secret is s_1 (for every two secrets s_0, s_1), and the scheme is secure. □

Remark 4.10 (Historical Notes)**.** Brickell [49] in 1989 implicitly defined monotone span programs for the case that each party labels exactly one row, and proved Claim 4.9. Karchmer and Wigderson [111] in 1993 explicitly defined span programs and monotone span programs. They considered them as a computational model and their motivation was proving lower bounds for modular branching programs. Karchmer and Wigderson showed that monotone span programs imply (linear) secret-sharing schemes. Beimel [14] proved that linear secret-sharing schemes (under various definitions) imply monotone span programs. Thus, linear secret-sharing schemes are equivalent to monotone span programs, and lower bounds on the size of monotone span programs imply the same lower bounds on the information ratio of linear secret-sharing schemes.

Example 4.11. We next describe the linear secret-sharing for Γ_{ustcon}, presented in Section 4.1, as a monotone span program over $\mathbb{F}_2$. Recall that in Γ_{ustcon} we consider a graph with m vertices and $n = \binom{m}{2}$ edges; each edge is a party. We construct a monotone span program over $\mathbb{F}_2$, which has $b = m - 1$ columns and $a = n$ rows. For each party (edge) (v_i, v_j), where $1 \leq i < j \leq m - 1$ (i.e., $j \neq m$), there is a unique row in the program labeled by this party; all entries in this row are zero, except for the ith and the jth entries, which are 1. For $j = m$ and for each party (edge) (v_i, v_m), where $1 \leq i \leq m - 1$, there is a unique row in the program labeled by this party; all entries in this row are zero, except for the ith entry, which is 1 (this is equivalent to choosing $r_m = 0$ in Section 4.1). It can be proved that this monotone span program accepts a set of parties (edges) if and only if the set contains a path from v_1 to v_m.

To construct a secret-sharing scheme from this monotone span program, we multiply the above matrix by a vector $\mathbf{r} = \langle s, r_2, \ldots, r_{m-1} \rangle$ and the share of party (v_i, v_j) is the row labeled by (v_i, v_j) in the matrix multiplied by $\mathbf{r}$. That is, the share is as defined in the scheme for Γ_{ustcon} described in Figure 4.1. For example, the share of edge (v_1, v_2) is $\langle 1, 1, 0, \ldots, 0 \rangle \langle s, r_2, \ldots, r_{m-1} \rangle^T = s \oplus r_2$.

Clearly, every secret-sharing scheme from a monotone span program (i.e., the scheme of Figure 4.6) is linear according to Definition 4.1. The converse is also true as shown in the next claim, which basically follows from the proof of Claim 4.9.

Claim 4.12. *Let q be a prime-power and $\ell_1, \ldots, \ell_n \in \mathbb{N}$. Assume that there is a linear secret-sharing scheme Π over $\mathbb{F}_q$ according to Definition 4.1 realizing an access structure Γ, where for every $1 \leq i \leq n$ the domain of shares of party p_i is $\mathbb{F}_q^{\ell_i}$. Then there is a monotone span program $\langle \mathbb{F}_q, M, \rho \rangle$ accepting Γ, whose size is $\sum_{i=1}^{n} \ell_i$.*

Proof. Recall that in a linear secret-sharing scheme each share is a vector over $\mathbb{F}_q$, where each coordinate is a linear combination of the secret and random elements from $\mathbb{F}_q$. Let M be the linear mapping of the sharing of the secret-sharing scheme Π, where the first column of M contains the coefficients of the secret in any

linear combination. In other words, the shares are computed by

$$M \begin{pmatrix} s \\ r_2 \\ \vdots \\ r_b \end{pmatrix}.$$

Furthermore, assume that the first ℓ_1 rows compute the share of p_1, the next ℓ_2 rows compute the share of p_2, and so on. For every $1 \leq i \leq n$ and every $1 + \sum_{m=1}^{i-1} \ell_i \leq j \leq \sum_{m=1}^{i} \ell_i$, let $\rho(j) = i$. We need to prove that the monotone span program accepts Γ. As proved in Claim 4.9, a set of parties A can reconstruct the secret in the scheme Π if and only if $\mathbf{e_1}$ is spanned by the rows of M_B. That is, $\langle \mathbb{F}_q, M, \rho \rangle$ accepts Γ as claimed. □

4.5 Properties of Linear Secret-Sharing Schemes

Linear secret-sharing schemes have interesting properties that are useful for many applications. We next list a few of them.

Linear Sharing vs. Linear Reconstruction. In the proof of Claim 4.9, we have shown that if we have a secret-sharing scheme in which the sharing is computed by a linear mapping, then the reconstruction is also computed by a linear mapping (see (4.1)). The converse is also true; it is proved in [14] that if we have a secret-sharing scheme where each authorized set reconstructs the secret by applying a linear mapping to its shares, then the scheme can be converted to an equivalent scheme with the same share size in which the sharing is computed by a linear mapping.

Efficiency of Sharing and Reconstruction. In linear secret-sharing schemes, both sharing and reconstruction are done by computing a linear transformation; thus, if the share size in the scheme is reasonable, the running times of the algorithms computing the sharing and reconstruction are efficient.

Zero-One Law for Security/Correctness. Let Π be a linear secret-sharing scheme Π, whose linear mapping is defined by a matrix M. For any set of parties A, either the rows of M_A span the vector $\mathbf{e_1}$, i.e., the parties in A can reconstruct the secret, or the rows of M_A do not span the vector $\mathbf{e_1}$, i.e., the parties in A do not learn any information on the secret. That is, in any linear secret-sharing scheme, there are no sets that can learn partial information on the secret. This property can be used to simplify proofs that a set T cannot learn any information in a secret-sharing scheme – it suffices to show that some shares of T can be generated for two secrets (e.g., 0 and 1).

Extending the Domain of Secrets. If we have a linear secret-sharing scheme over some finite field $\mathbb{F}_q$ with a domain of secrets $S \subsetneq \mathbb{F}_q$, then we can extend the secret-sharing scheme to a secret-sharing realizing the

same access structure where the domain of secrets is $\mathbb{F}_q$ and the domain of shares of each party is not changed. In other words, we do not pay in the share size for extending the domain of secrets.

To see this, suppose we have a secret-sharing scheme Π realizing Γ such that Π is linear over $\mathbb{F}_q$ as defined in Definition 4.1, where the domain of secrets is $S \subsetneq \mathbb{F}_q$ of size at least 2, i.e., $s_0, s_1 \in S$ for some $s_0, s_1 \in \mathbb{F}_q$. Let M be the linear mapping computing the sharing function in Π and ρ be the labeling of the rows by parties. We claim that, by the proof of Claim 4.9, although the domain of secrets is not the entire field, Γ is the access structure accepted by the monotone span program $\mathsf{MSP} = \left\langle \mathbb{F}_q, M, \rho \right\rangle$. I.e., if the rows of M_A span $\mathbf{e_1}$ then by the proof of Claim 4.9 the parties in A can reconstruct the secret in Π and $A \in \Gamma$ and if the rows of M_T do not span $\mathbf{e_1}$, then for every vector of shares $\left\langle \mathsf{sh}_j \right\rangle_{p_j \in T}$ the probability that the shares are generated in Π with the secrets s_0, s_1 is the same and $T \notin \Gamma$. Thus, by Claim 4.9, we can use MSP to realize Γ with the entire field $\mathbb{F}_q$ as the domain of secrets and the same domain of shares.

Additivity. If we take shares in a linear secret-sharing scheme and add them, then we get shares of the sum of the two secrets. Formally,

Lemma 4.13. *Let $s_1, s_2 \in \mathbb{F}_q$ be two secrets and for $b \in \{1,2\}$ let $\mathsf{sh}_1^b, \ldots, \mathsf{sh}_n^b$ be shares of the secret s_b in a linear secret-sharing scheme for a monotone span program $\mathsf{MSP} = \left\langle \mathbb{F}_q, M, \rho \right\rangle$, where $s_j^b = \left\langle s_{j,1}^b, \ldots, s_{j,a_j}^b \right\rangle \in \mathbb{F}_q^{a_j}$ for some integer a_j. Define $\mathsf{sh}_j = \left\langle s_{j,1}^1 + s_{j,2}^1, \ldots, s_{j,\ell_j}^1 + s_{j,a_j}^2 \right\rangle$. Then, $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$ are shares of $s_0 + s_1$.*

Proof. Without loss of generality, assume that the first a_1 rows of M are labeled by p_1, the next a_2 rows are labeled by p_2 and so on. Let $\mathbf{r_1} = \left\langle s_1, r_2^1, \ldots, r_b^1 \right\rangle$ and $\mathbf{r_2} = \left\langle s_2, r_2^2, \ldots, r_b^2 \right\rangle$ be random strings such that $\left\langle \mathsf{sh}_1^1, \ldots, \mathsf{sh}_n^1 \right\rangle = M\mathbf{r_1}$ and $\left\langle \mathsf{sh}_1^2, \ldots, \mathsf{sh}_n^2 \right\rangle = M\mathbf{r_2}$. Thus,

$$M(\mathbf{r_1} + \mathbf{r_2}) = M\mathbf{r_1} + M\mathbf{r_2} = \left\langle \mathsf{sh}_1^1, \ldots, \mathsf{sh}_n^2 \right\rangle + \left\langle \mathsf{sh}_1^2, \ldots, \mathsf{sh}_n^2 \right\rangle = \langle \mathsf{sh}_1, \ldots, \mathsf{sh}_n \rangle .$$

As the first coordinate in $\mathbf{r_1} + \mathbf{r_2}$ is $s_1 + s_2$, the secret $s_1 + s_2$ and randomness $r_2^1 + r_2^2, \ldots, r_b^1 + r_b^2$ generate the shares $\langle \mathsf{sh}_1, \ldots, \mathsf{sh}_n \rangle$. □

Protocols for Multiplication. The additivity of linear secret-sharing schemes enables parties to compute, without any interaction, shares of the sum of two shared secrets. We would like to enable parties to compute shares of a product of two shared secrets, without revealing any information on the secret. This property will be useful when designing secure multiparty protocols (see Chapter 6). Unlike addition, this task requires interaction between the parties and can be carried out only if the union of any two unauthorized sets in the access structure is not the entire set of parties (e.g., in a t-out-of-n access structure, this is possible only if $2t - 1 < n$); this property is called Q^2. Cramer et al. [64] showed that if an n-party Q^2 access structure has a *linear* secret-sharing with total share size a, then there is a linear secret-sharing scheme with total share size $2a$ that has a secure 1-round protocol for computing the shares of a product of two shared secrets.

Duality. The dual of an access structure Γ with a set of parties P is the access structure

$$\Gamma^{\perp} = \{B : P \setminus B \notin \Gamma\},$$

that is, a set is authorized in the dual access structure $\Gamma^{\perp}$ if and only if its dual is unauthorized in Γ. For example, consider the t-out-of-n access structure $\Gamma_{n,t} = \left\{A \subseteq \left\{p_1, \ldots, p_n\right\} : |A| \geq t\right\}$; its dual is the $(n-t+1)$-out-of-n access structure $\Gamma^{\perp}_{n,t} = \Gamma_{n,n-t+1} = \left\{B \subseteq \left\{p_1, \ldots, p_n\right\} : |B| \geq n-t+1\right\}$.. If an access structure Γ has a linear secret-sharing scheme over $\mathbb{F}_q$ with information ratio a, then $\Gamma^{\perp}$ also has a linear secret-sharing scheme over $\mathbb{F}_q$ with information ratio a. For a proof of this claim see [86].[12]

Limitations of Linear Secret-Sharing Schemes. As explained, the access structures that can efficiently be realized by linear secret-sharing schemes are characterized by functions that have polynomial size monotone span programs. We will show that this implies that only access structures that can be represented by (non-monotone) NC circuits (i.e., by a Boolean circuit with polynomial number of gates and poly-logarithmic depth). To decide if a set A is accepted by a monotone span program $\left\langle \mathbb{F}_q, M, \rho \right\rangle$, we need to check if $\mathbf{e_1}$ is spanned by the rows of M_A, i.e., if $\text{rank}_{\mathbb{F}_q}(M_A) = \text{rank}_{\mathbb{F}_q}(M_A \cup \left\{\mathbf{e_1}\right\})$. By [138], computing the rank over $\mathbb{F}_q$ of an $n \times n$ matrix can be done by a polynomial size circuit of depth $O(\log^2(n) \log\log(q))$. Thus, access structures that cannot be represented by an NC circuit do not have an efficient linear secret-sharing scheme. For example, if $\mathsf{P} \neq \mathsf{NC}$, then access structures recognized by monotone P-complete problems do not have efficient linear secret-sharing schemes.

4.6 Multilinear Secret-Sharing Schemes

In the schemes derived from monotone span programs, the secret is one element from the field. This can be generalized to the case where the secret is some vector over the field. Such schemes, studied by [39, 41, 76], are called multilinear;[13] they are based on the following generalization of monotone span programs.

Definition 4.14 (Multi-Target Monotone Span Programs)**.** *A multi-target monotone span program is a quadruple* $\mathsf{MSP} = \langle \mathbb{F}, M, \rho, V \rangle$*, where* $\mathbb{F}$ *is a finite field,* M *is an* $a \times b$ *matrix over* $\mathbb{F}$*,* $\rho : \{1, \ldots, a\} \to \left\{p_1, \ldots, p_n\right\}$ *labels each row of* M *by a party, and* $V = \left\{\mathbf{e_1}, \ldots, \mathbf{e_c}\right\}$ *is a set of vectors in* $\mathbb{F}^b$ *for some* $1 \leq c < b$ *such that for every* $A \subseteq \left\{p_1, \ldots, p_n\right\}$ *either*

- *The rows of* M_A *span each vector in* $\left\{\mathbf{e_1}, \ldots, \mathbf{e_c}\right\}$*; in this case, we say that* MSP *accepts* A*, or,*

[12] It is not known if general secret-sharing schemes have this property, i.e., it is not known if an access structure can be realized by a secret-sharing scheme with information ratio a implies that $\Gamma^{\perp}$ has a secret-sharing scheme with information ratio $O(a)$ or $\text{poly}(a)$. See [71] for a discussion on this subject and some separation between the share size of an access structure and its dual. See also [46] for an example of access structures closed under duality.

[13] The name multilinear secret-sharing scheme is inconsistent with the notion of multilinear maps (i.e., maps that are linear in each variable but the mapping itself can have larger degree). It borrows its name from multilinear representable matroids (e.g., in [167]); to be consistent with the literature on secret-sharing schemes we will use this name in this monograph.

- *The rows of M_A span no non-zero vector in the linear space spanned by $\left\{\mathbf{e_1}, \ldots, \mathbf{e_c}\right\}$; in this case, we say that* MSP *rejects A.*

We say that MSP *accepts an access structure Γ if* MSP *accepts a set B iff $B \in \Gamma$ and rejects a set B iff $B \notin \Gamma$.*

Not that, in general, it is possible that M neither accepts some set A nor rejects it. In this case, $\langle \mathbb{F}, M, \rho, V \rangle$ is not a multi-target MSP.

Claim 4.15. *Let $\mathsf{MSP} = \left\langle \mathbb{F}_q, M, \rho, V \right\rangle$ be a multi-target monotone span program accepting Γ, where $\mathbb{F}_q$ is a finite field, $|V| = c$, and for every $j \in \{1, \ldots, n\}$ there are a_j rows of M labeled by p_j. Then, there is a multilinear secret-sharing scheme realizing Γ such that the secret is a vector in $\mathbb{F}_q^c$ and the share of party p_j is a vector in $\mathbb{F}_q^{a_j}$; in particular, the information ratio of the scheme is $\max_{1 \leq j \leq n} a_j / c$.*

The proof of Claim 4.15 is similar to the proof of Claim 4.9, where in this case the secret is $s_1, \ldots, s_c$, the dealer chooses $b - c$ random elements $r_{c+1}, \ldots, r_b$ in $\mathbb{F}_q$, uses the vector $\mathbf{r} = \left\langle s_1, \ldots, s_c, r_{c+1}, \ldots, r_b \right\rangle$, and computes the shares $M\mathbf{r}$. The correctness is similar to the proof of Claim 4.9, i.e., if $\mathbf{e_i}$ is spanned by the rows of M_A, then the parties in A can reconstruct s_i. The security follows from the fact that if the rows of M_T do not span any non-trivial linear combination of $\mathbf{e_1}, \ldots, \mathbf{e_c}$, then for every $1 \leq i \leq c$, the rows of M_T and $\mathbf{e_1}, \ldots, \mathbf{e_{i-1}}, \mathbf{e_{i+1}}, \ldots, \mathbf{e_c}$ do not span $\mathbf{e_i}$, i.e., there exists a vector $\mathbf{w}$ such that $(M_T)\mathbf{w} = \mathbf{0}$, $\mathbf{e_j} \cdot \mathbf{w} = 0$ (i.e., $w_j = 0$) for every $j \neq i$, and $\mathbf{e_i} \cdot \mathbf{w} = 1$ (i.e., $w_i = 1$). As in the proof of Claim 4.9, this implies that for every two secrets that differ only in the i-th coordinate and for every possible vector of shares for T, the probability that the shares are generated for the two secrets is the same. By applying the above arguments at most c times, we get the security for every two secrets.

Any multi-target monotone span program is a monotone span program; however, using multi-target monotone span programs to construct a multilinear secret-sharing scheme results in a scheme with a better information ratio.

Example 4.16. Consider the access structure $\Gamma_{\sqcap}$ defined in Example 3.2. It was proved by Capocelli et al. [53] that in any secret-sharing scheme realizing $\Gamma_{\sqcap}$ the information ratio is at least 1.5. We present this lower bound and prove it in Theorem 7.4. By definition, the information ratio of a linear scheme is integral; thus, in any linear secret-sharing realizing $\Gamma_{\sqcap}$ the information ratio is at least 2.

We next present a multilinear secret-sharing scheme realizing $\Gamma_{\sqcap}$ with information ratio 1.5. In Example 4.2 we described a linear scheme realizing $\Gamma_{\sqcap}$ whose information ratio is 2; this scheme is the BL scheme for the formula $\big((x_1 \vee x_3) \wedge x_2\big) \vee (x_3 \wedge x_4)$. Notice that although parties p_2 and p_3 have symmetric roles in $\Gamma_{\sqcap}$, in the scheme described Example 4.2 party p_2 gets one field element and p_3 gets two field elements. To construct a multilinear scheme realizing $\Gamma_{\sqcap}$ whose information ratio is 1.5, we exploit the asymmetry of the previous scheme. To share a secret $\langle s_1, s_2 \rangle \in \mathbb{F}_q^2$, the dealer shares s_1 as above and shares s_2 using the BL scheme for the formula $\big(x_2 \vee x_4) \wedge x_3\big) \vee (x_1 \wedge x_2)$. The scheme is described in the following table, where r_1, r_2, r_3, and r_4 are uniformly distributed random elements in $\mathbb{F}_q$.

Secret	Randomness	Share of p_1	Share of p_2	Share of p_3	Share of p_4
$s_1, s_2 \in \mathbb{F}_q$	$r_1, r_2, r_3, r_4 \in \mathbb{F}_q$	r_1, r_3	$r_1 \oplus s_1, r_3 \oplus s_2, r_4$	$r_1, r_2 \oplus s_1, r_4 \oplus s_2$	r_2, r_4

The secret in the above scheme is two field elements and the largest shares are 3 elements; hence the information ratio of this scheme (i.e., the ratio between the share size and the secret size) is 1.5. It is an easy exercise to write the above multilinear scheme as a multi-target monotone span program; the matrix of this program has 10 rows and 6 columns.

The scheme in Example 4.16 involves two applications of linear secret-sharing schemes realizing $\Gamma_{\sqcap}$, each application with an independent secret and independent random bits. In particular, the multilinear secret-sharing scheme has the same total information ratio as the linear scheme. Multilinear secret-sharing schemes are provably more efficient than linear secret-sharing schemes [167, 5, 4, 19]. Specifically, Applebaum and Arkis [4] constructed, for every n, multilinear secret-sharing schemes with information ratio 4 for a family of $2^{2^{n/2}}$ access structures (alas for long secrets of size $2^{n^{n/2}}$); this family is the family of all $n/2$-partite access structures.[14] By [19] the number of access structures that can be realized by a linear secret-sharing scheme (over all fields) with information ratio d is at most $0.5 \cdot 2^{d^3/2}$. Thus, for almost all $n/2$-partite access structures, the information ratio of every linear secret-sharing realizing them is at least $2^{n/6}$.

[14] In an $n/2$-partite access structure, the n parties are partitioned to $n/2$ pairs, $\{p_1, p_2\}, \{p_3, p_4\}, \ldots, \{p_{n-1}, p_n\}$. The authorized sets are all sets of size at least $n/2 + 1$ and some sets of size $n/2$ containing exactly one party from each pair.

Chapter 5

Secret-Sharing Schemes for Arbitrary Access Structures with Exponent Smaller Than One

In a breakthrough paper, Liu and Vaikuntanathan [123] constructed for every access structure a secret-sharing scheme with share size $2^{0.994n}$. This was improved in a sequence of works [6, 8, 9] to share size $(3/2)^{(1+o(1))n} < 2^{0.585n}$ [9]. These constructions are rather involved and, in particular, use constructions of matching vectors [102]. In this chapter we describe a simpler construction, from [8], of a secret-sharing scheme with share size 2^{cn} for some constant $0 < c < 1$; for the sake of simplicity, we do not try to optimize this constant. Our description will be self-contained and contain all details (except for a few simple claims). As a bonus, the scheme we will describe is linear (in [123, 6, 8, 9, 2], linear schemes are also described).

To construct the scheme, in Section 5.1 we define robust graph secret-sharing schemes and, as a warm-up and a motivation for this definition, we show that if there is a fully-robust graph secret-sharing scheme, then we can construct a secret-sharing scheme for an arbitrary access structure. However, if we plug in the best known fully robust graph secret-sharing scheme, the share size of the resulting secret-sharing scheme would be $2^{n-o(n)}$. In the end of this chapter, in Section 5.4, we show a more complicated reduction from secret-sharing schemes for arbitrary n-party access structure to a t-robust graph secret-sharing scheme for a graph with $N = O(2^{n/2})$ vertices and $t \ll N$. To complete the construction, we describe in Section 5.2 a 1-robust graph secret-sharing scheme and in Section 5.3 a transformation of this scheme to a t-robust graph secret-sharing scheme.

5.1 Robust Graph Secret Sharing

In a graph secret-sharing scheme [50], the parties are vertices of a graph and a set of vertices (parties) can reconstruct the secret if and only if it contains an edge. In other words, all minimal authorized sets are of size 2 and a set is unauthorized if it is an independent set in the graph. Graph secret-sharing schemes

were studied in many papers, e.g., [50, 51, 53, 77, 43, 68, 73, 69, 70, 21, 82, 72]. The naive scheme to realize a graph is to share the secret independently for each edge; this result implies a share of size $O(N)$ per party for an N-vertex graph. A better scheme with share size $O(N/\log(N))$ per party is implied by a result of Erdös and Pyber [80]. For constructing secret-sharing schemes for arbitrary access structures, we will need the following generalization of this notion. In a t-robust graph secret-sharing scheme [8, 19], a set of vertices (parties) can reconstruct the secret if and only if it contains an edge *or if the set contains at least $t+1$ vertices*. Using this terminology, N-robust graph secret-sharing schemes (for a non-empty graph) are graph secret-sharing schemes. In the other extreme, 2-robust graph secret-sharing schemes are called forbidden graph secret-sharing schemes [170] and are basically equivalent to 2-server conditional disclosure of secrets (CDS) protocols [95]. For every N-vertex graph there is a linear 2-robust graph secret-sharing scheme with share size $\tilde{O}(N^{1/2})$ [93] and a non-linear 2-robust graph secret-sharing scheme with share size $N^{\tilde{O}(1/\sqrt{\log(N)})} = 2^{\tilde{O}(\sqrt{\log(N)})}$ [125]. In this monograph, we define robust secret-sharing schemes only for bipartite graphs, as this is the graphs we need. For our construction, we will allow different sizes of authorized sets from each part of the graph.

Definition 5.1. *Let $G = (U, V, E)$ be a bipartite undirected graph (i.e., $E \subseteq U \times V$) and t_1, t_2 be integers. Define the (t_1, t_2)-robust graph access structure Γ_{G,t_1,t_2}, where the parties in the access structure are the vertices in $U \cup V$ and a set $A \subseteq U \cup V$ is authorized if at least one of the following conditions holds:*

- *The set A contains at least one edge, i.e., there are $u, v \in A$ such that $(u, v) \in E$,*
- *$|A \cap U| > t_1$, or*
- *$|A \cap V| > t_2$.*

A (t_1, t_2)-robust secret-sharing scheme for the graph G is a secret-sharing scheme realizing the access structure Γ_{G,t_1,t_2}.

We next describe a simple reduction from realizing an arbitrary access structure to (fully-robust) graph secret sharing. Given an access structure Γ with parties $p_1, \ldots, p_n$ we define the following bipartite graph $G = (U, V, E)$ with N vertices, where $N = 2 \cdot 2^{n/2}$:

- $U = 2^{\{p_1,\ldots,p_{n/2}\}}$, i.e., the vertices in the left side are the subsets of $\{p_1, \ldots, p_{n/2}\}$.
- $V = 2^{\{p_{n/2+1},\ldots,p_n\}}$.[15]
- For every minimal authorized set A in Γ, there is an edge

$$\left(A \cap \{p_1, \ldots, p_{n/2}\}, A \cap \{p_{n/2}+1, \ldots, p_n\}\right)$$

  in E.

An illustration of a construction of such a graph appears in Figure 5.1.

The secret-sharing for Γ is as follows:

[15] There is vertex for the empty set in both sides. These are two different vertices.

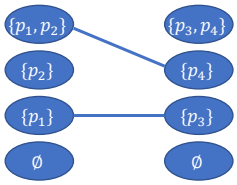


Figure 5.1: The graph G constructed from the access structure with two minimal authorized sets $\{p_1, p_3\}$ and $\{p_1, p_2, p_4\}$.

- Construct the above graph $G = (U, V, V)$ for the access structure Γ.
- Share the secret s using any fully-robust secret-sharing scheme for G. Let sh_C be the share in this scheme of the vertex $C \in U \cup V$.
- For every non-empty set $C \in U \cup V$ (where C is a set of parties), independently share sh_C using the $|C|$-out-of-$|C|$ secret-sharing scheme of Example 1.1 among the parties of C. In addition, give the shares of $\emptyset \in U$ and $\emptyset \in V$ to all parties.

We next argue the correctness and security of the scheme. First, let $A = A_1 \cup A_2$ be a minimal authorized set in Γ, where $A_1 \subseteq \{p_1, \ldots, p_{n/2}\}$ and $A_2 \subseteq \{p_{n/2+1}, \ldots, p_n\}$. By the construction of G, the edge (A_1, A_2) is in E, thus sh_{A_1} and sh_{A_2} determine the secret. Furthermore, the parties in A can reconstruct sh_{A_1} and sh_{A_2}, hence, can reconstruct the secret.

For the security of the scheme, consider an unauthorized set $T = T_1 \cup T_2 \notin \Gamma$, where $T_1 \subseteq \{p_1, \ldots, p_{n/2}\}$ and $T_2 \subseteq \{p_{n/2+1}, \ldots, p_n\}$. Clearly, the parties in T can reconstruct sh_{T_1} and sh_{T_2}; however, they can also reconstruct the shares of every subset of T_1 and every subset of T_2. On the other hand, for any other vertex in $B \in U \cup V$, the parties in T miss at least one party in B. Hence, the parties in T have no information of the shares of these vertices.

Since T is unauthorized, every subset of T is unauthorized and there are no edges between subsets of T_1 and subsets of T_2, i.e., the parties in T hold shares of an independent set in G. By the fully-robustness of the scheme for G, the shares that the parties in T hold do not give any information on the secret s.

Example 5.2. In the graph described in Figure 5.1, the unauthorized set $\{p_2, p_3, p_4\}$ can reconstruct $\mathsf{sh}_{\{p_2\}}, \mathsf{sh}_\emptyset$ from the left side and the shares of all vertices from the right side. There is no edge between the vertices $\{p_2\}, \emptyset$ and the vertices in the right side, so these shares give no information on the secret.

We next analyze the share size in the above scheme. Party p_i gets a share of sh_A for every A such that $p_i \in A$; there are $2^{n/2-1}$ such sets. Thus, the share size of p_i is $O(2^{n/2-1} \cdot \max_{p_i \in A} |\mathsf{sh}_A|)$. The share size in the best known fully-robust secret-sharing scheme for an $O(N)$-vertex graph is $O(N/\log(N))$ [80]. Recalling that the graph G has $O(2^{n/2})$ vertices, the best known implementation of the above scheme has share size $2^{n-o(n)}$. We do not know if there is a fully-robust graph secret-sharing scheme with share size $O(N^c)$ for some $c < 1$. We bypass this by using t-robust graph secret-sharing schemes for $t \ll N$.

In Section 5.4, we will show a more complicated reduction to t-robust graph secret-sharing schemes for $t \ll 2^{n/2}$. The high-level idea of this reduction is to ensure that $T_1 = T \cap \{p_1, \ldots, p_{n/2}\}$ is small for every unauthorized set T. As we only need that $\{A_1 \subseteq T \cap U\} \cup \{A_2 \subseteq T \cap V\}$ is unauthorized for every $T \notin \Gamma$, the robustness that we need, i.e., the number of subsets of $T \cap U$, is small.

5.2 A $(1, N)$-Robust Graph Secret-Sharing Scheme

In Figure 5.2), we describe a $(1, N)$-robust graph secret-sharing scheme $\Pi_{\text{OneRobust}}$ for a bipartite graph $G = (U, V, E)$, i.e., a scheme in which a pair of parties can reconstruct the secret if and only if they are connected by an edge and every set of size 3 can reconstruct the secret. This scheme is a variant of a scheme of [93]. In this scheme there is a random bit r_u for every $u \in U$, the share of u are all these random bits *except for* r_u and the share of v is the exclusive-or of the secret and the random bits of the non-neighbors of v. Every non-edge u, v does not learn any information on the secret as u does not hold r_u and r_u masks the secret in the share of v. This scheme is not $(2, 1)$-robust as every two vertices in U hold all of the bits r_u, thus together with any non-neighbor vertex in V they can reconstruct the secret. However, we will show that it is $(1, N)$-robust.

Scheme $\Pi_{\text{OneRobust}}$

The secret: A bit $s \in \{0, 1\}$.
The scheme:

1. Choose $|U| + 1$ random bits $r_0, \langle r_u \rangle_{u \in U}$.
2. The share of a vertex $u \in U$ is $\mathsf{sh}_u = \left\langle r_0, \langle r_w \rangle_{w \in U, w \neq u} \right\rangle$ (that is, all random bits except for r_u).
3. The share of a vertex $v \in V$ is $\mathsf{sh}_v = s \oplus r_0 \oplus \bigoplus_{u \in U, (u,v) \notin E} r_u$.

Figure 5.2: A $(1, |V|)$-robust graph secret-sharing $\Pi_{\text{OneRobust}}$ for a bipartite graph $G = (U, V, E)$.

Lemma 5.3. *Let $G = (U, V, E)$ be a bipartite graph. Then, the scheme $\Pi_{\text{OneRobust}}$, described in Figure 5.2, is a $(1, |V|)$-robust graph secret-sharing scheme for G in which the share size of each vertex in U is $|U|$ and the share size of each vertex in V is 1.*

Proof. For the correctness of the scheme $\Pi_{\text{OneRobust}}$, consider an edge $(u, v) \in E$. In this case, r_u is not part of the exclusive-or in the share sh_v held by v and the vertex u holds all random bits except for r_u. Thus, u and v can reconstruct the secret.

For the $(1, |V|)$ robustness of the scheme, consider a set $T_2 \subseteq V$ and a vertex u such that $(u, v) \notin E$ for every $v \in T_2$ (i.e., $\{u\} \cup T_2$ is unauthorized). Note that $s \oplus r_u$ appears as a term in sh_v for every $v \in T_2$ and

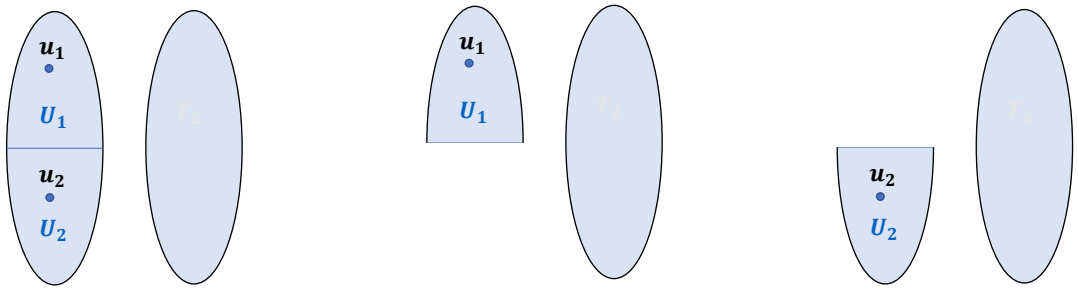
r_u is not part of the share of u. Thus, r_u acts as a one-time pad and completely hides s. □

Remark 5.4. In $\Pi_{\text{OneRobust}}$, the share of a vertex u is "big", while the share of a vertex $v \in V$ is a bit. We can balance the share size; e.g., when $|U| = |V| = N$, we can construct a $(1, N)$-robust secret-sharing scheme with share size $O(N^{1/2})$ for every vertex. That is, assuming $|U| = |V| = N$, we partition U to $\sqrt{N}$ sets $U_1, \ldots, U_{\sqrt{N}}$ of size $\sqrt{N}$ and execute $\Pi_{\text{OneRobust}}$ for each graph $G_i = (U_i, V, E \cap (U_i \times V))$. Each party in U participates in one execution of $\Pi_{\text{OneRobust}}$, with share size $|U_i| = \sqrt{N}$ and each party in V participates in $\sqrt{N}$ executions of $\Pi_{\text{OneRobust}}$, each execution with share size 1. We will implicitly use this balancing when constructing a robust secret-sharing scheme from $\Pi_{\text{OneRobust}}$.

5.3 A (t, N)-Robust Graph Secret-Sharing Scheme

Next, we show how to transform the scheme $\Pi_{\text{OneRobust}}$ to a (t, N)-robust secret-sharing scheme Π_{Robust} for $N^{1/4} \leq t \leq N^{1/2}$.

Warm Up: $(2, N)$-Robust Secret Sharing. Suppose we have an unauthorized set $T = \{u_1, u_2\} \cup T_2$. We randomly partition the set U into two sets U_1, U_2 and independently share the secret s in a $(1, N)$-robust secret-sharing scheme for the graph $G_1 = (U_1, V, E \cap (U_1 \times V))$ and for the graph $G_2 = (U_2, V, E \cap (U_2 \times V))$. If $u_1 \in U_1$ and $u_2 \in U_2$, then, by the $(1, N)$-robustness of the schemes for G_1 and for G_2, the parties in T do not learn any information on s from the scheme for G_1 or from the scheme for G_2. An illustration of the partition appears in Figure 5.3. However, if we are unlucky and u_1, u_2 are in the same set U_i, there are no guarantees.



The graph G The graph G_1 The graph G_2

Figure 5.3: The partition of the graph to two graphs.

To overcome this problem, we use $\ell = 2\log(N)$ random partitions $\left\langle U_1^j, U_2^j \right\rangle_{1 \leq j \leq \ell}$, where for each j

we independently choose U_1^j with uniform distribution and define $U_2^j = U \setminus U_1^j$. We share the secret s as follows:

1. Share s using the ℓ-out-of-ℓ secret-sharing scheme of Example 1.1 to produce shares $\mathsf{sh}_1, \ldots, \mathsf{sh}_\ell$.
2. For $j = 1$ to ℓ do:
   - Independently share sh_j using the $(1, N)$-robust secret-sharing scheme $\Pi_{\text{OneRobust}}$ for the graph $G_1^j = (U_1^j, V, E \cap (U_1^j \times V))$ and for the graph $G_2^j = (U_2^j, V, E \cap (U_2^j \times V))$.

All together, there are 2ℓ executions of $\Pi_{\text{OneRobust}}$. If we take the secret-sharing scheme discussed in Remark 5.4 (instead of $\Pi_{\text{OneRobust}}$), then the share size in this scheme is $O(N^{1/2}\log(N))$.

The correctness for an edge $(u, v) \in E$ is immediate as for every j there is an index $i_j \in \{1, 2\}$ such that $u \in U_{i_j}^j$ and u, v can reconstruct sh_j from the scheme for $G_{i_j}^j$. For the $(2, N)$-robustness, fix two vertices u_1, u_2. If for at least one j, the vertices u_1, u_2 are in different sets U_1^j, U_2^j, then for every unauthorized set $T = \{u_1, u_2\} \cup T_2$ the parties in T cannot learn any information on sh_j, hence they cannot learn any information on s. The probability that such j exists for u_1, u_2 is $1 - 1/2^\ell = 1 - 1/N^2$. By a simple probabilistic argument, there exist $\ell = 2\log(N)$ partitions $\left\langle U_1^j, U_2^j \right\rangle_{1 \le j \le \ell}$ such that for every u, v there exists an index j such that $|U_1^j \cap \{u, v\}| = |U_2^j \cap \{u, v\}| = 1$.[16] In this case, the $(2, N)$-robustness follows.

To get (t, N)-robustness, we use the same process, i.e., we partition the sets U to sets $U_1, \ldots, U_m$ such that for a set $T_1 \subseteq U$ of size t it would hold that $|T_1 \cap U_i| \le 1$ for every $1 \le i \le m$. To ensure that we can use few partitions, we take $m = O(t^2)$.[17]

The scheme Π_{Robust} is described in Figure 5.4, where $\ell = O(t\log(N))$ and $\left\langle (U_1^j, \ldots, U_{t^2}^j) \right\rangle_{1 \le j \le \ell}$ is a sequence of partitions satisfying the following two requirements:

- For every $T_1 \subseteq U$ of size t there exists at least one index j such that $|U_h^j \cap T_1| \le 1$ for every $1 \le h \le t^2$, that is, each set U_h^j contains at most one party from T_1.
- For every i, j, $|U_i^j| \le \left\lceil |U|/t^2 \right\rceil$ (that is, the sizes of the sets $U_1^j, \ldots, U_{t^2}^j$ in each partition are almost equal).

By a simple probabilistic proof, such sequence exists.

Lemma 5.5. *Let $G = (U, V, E)$ be a bipartite graph such that $|U|, |V| \le N$, and $|U|^{1/4} \le t \le |U|^{1/2}$ be an integer. Then, the scheme Π_{Robust} is a (t, N)-robust secret-sharing scheme for G with one-bit secrets in which the share size of each party is $O(t^3\log(N))$.*

[16] For this warm up case of $(2, N)$-robustness, there is a simple explicit construction of $\log(N)$ partitions satisfying the above property.

[17] If we take, for example, $m = O(t\log(N))$, then we will need $\Omega(2^t)$ partitions, which will result in an inefficient secret-sharing scheme.

Scheme Π_{Robust}

The secret: A bit $s \in \{0,1\}$.

The scheme:

1. Let $\left\langle (U_1^j, \ldots, U_{t^2}^j) \right\rangle_{1 \le j \le \ell = O(t \log(N))}$, where each $(U_1^j, \ldots, U_{t^2}^j)$ is a partitions of U to t^2 sets.
2. Share s using the ℓ-out-of-ℓ secret-sharing scheme of Example 1.1 to produce shares $\mathsf{sh}_1, \ldots, \mathsf{sh}_\ell$.
3. For $j = 1$ to ℓ do:
   - For every $1 \le i \le t^2$, independently share the secret sh_j using the $(1, N)$-robust secret-sharing scheme $\Pi_{\text{OneRobust}}$ for the graph $G_i^j = (U_i^j, V, E \cap (U_i^j \times V))$.

   ($*$ The scheme $\Pi_{\text{OneRobust}}$ is executed $t^2\ell = O(t^3 \log(N))$ times $*$)

Figure 5.4: A (t, N)-robust secret-sharing scheme Π_{Robust} for a bipartite graph $G = (U, V, E)$ with share size $O(t^3 \log(N))$ for $N^{1/4} \le t \le N^{1/3}$.

Proof. For the correctness of the scheme, consider $(u, v) \in E$. Then, for every $1 \le j \le \ell$, the edge (u, v) is in one graph G_i^j, thus u, v can reconstruct each sh_j, hence reconstruct s.

For the robustness of the protocol, let T_1, T_2 be sets such that $T_1 \subseteq U$, $T_2 \subseteq V$, $|T_1| \le t$, $|T_2| \le N$, and there is no edge $(u, v) \in E$ for $u \in T_1, v \in T_2$ (that is, $T_1 \cup T_2$ is unauthorized). By the above requirements on the sequence of partitions, there is at least one $1 \le j \le \ell$ for which $|U_h^j \cap T_1| \le 1$ for every $1 \le h \le t^2$. That is, for every i the vertices in $T_1 \cup T_2$ get at most one share of a vertex in U in the $(1, N)$-robust scheme for G_i^j, thus do not learn any information on sh_j from this scheme. Since all schemes are executed independently, the vertices in $T_1 \cup T_2$ do not learn any information on sh_j, and, thus, do not learn any information on the secret s.

We next provide an analysis of the share size. The share of a vertex $v \in V$ is composed of $t^2\ell = O(t^3 \log(N))$ shares of v in a $(1, N)$-robust secret-sharing scheme, each share is 1 bit. For every $1 \le j \le \ell$, a vertex $u \in U$ participates in a $(1, N)$-robust secret-sharing scheme for a graph G_i^j for exactly one i; the share size of u in this scheme is $|U_i^j| = O(|U|/t^2)$. Thus, the share size of u in is $O(\ell |U|/t^2) \le O(N \log(N)/t) \le O(t^3 \log(N))$ (where the last inequality follows from the fact that $t > N^{1/4}$). □

5.4 Secret Sharing Scheme from a Robust Secret Sharing

In this section we describe a reduction from realizing an arbitrary access structure to constructing a (t, N)-robust graph secret-sharing scheme. The reduction is similar to the construction in Section 5.1; however,

before constructing the bipartite graph we take 3 steps. See Figure 5.5 for a description of the formula describing these steps. We first remove small authorized sets and big unauthorized sets from the access structure (Γ_{BOT} and Γ_{TOP} in the figure). We then show that we can assume that every *minimal authorized set* contains exactly half of its parties in the set $\{p_1, \ldots, p_{n/2}\}$ (the OR between $\Gamma_{\text{MID},B_1}, \ldots, \Gamma_{\text{MID},B_w}$ in the figure). Finally, we show that we can assume that every *maximal unauthorized set* contains at most half of its parties in the set $\{p_1, \ldots, p_{n/2}\}$ (this is done inside each Γ_{MID,B_i} in the figure). These 3 steps will allow us to bound the number of subsets of an unauthorized set, which as explained in Section 5.1 bounds the required robustness t.

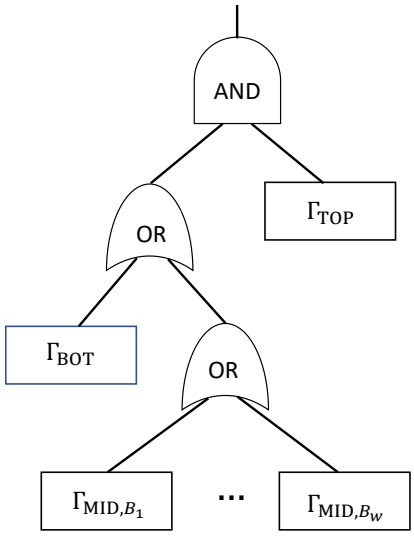


Figure 5.5: The formula describing the construction of the secret-sharing scheme for an arbitrary access structure Γ; the access structures $\Gamma_{\text{BOT}}, \Gamma_{\text{TOP}}, \Gamma_{\text{MID},B_1}, \ldots, \Gamma_{\text{MID},B_w}$ are described in Definitions 5.6 and 5.9 and can be realized by secret-sharing schemes with share size $2^{(0.996+o(1))n}$. Using the construction of [36] (see Section 4.3) this implies a secret-sharing for Γ with share size $2^{(0.996+o(1))n}$.

5.4.1 Liu and Vaikuntanathan's Decomposition of Access Structures

As in [123], we decompose in Definition 5.6 an access structure Γ to three parts: A bottom part Γ_{BOT}, which handles small authorized sets, a middle part Γ_{MID}, which handles medium-size authorized sets, and a top part Γ_{TOP}, which handles large unauthorized sets; these access structures are defined using some constants, which are chosen to guarantee share size 2^{cn} for some $c < 1$. The decomposition is illustrated in Figure 5.6. In Claim 5.7 we express Γ using this decomposition; the proof of the claim follows from a simple case analysis.

Definition 5.6. *Let Γ be an n-party access structure. Define the following access structures $\Gamma_{\text{TOP}}, \Gamma_{\text{BOT}}$, and*

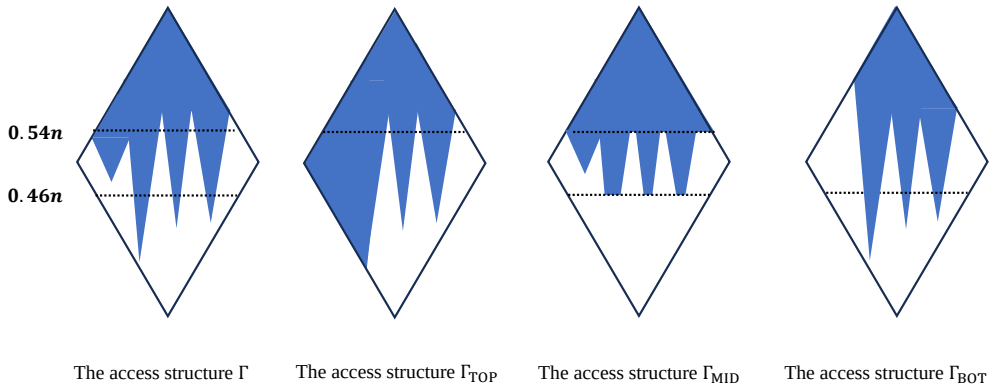


Figure 5.6: An example of the decomposition of Γ to $\Gamma_{\text{TOP}}, \Gamma_{\text{MID}}, \Gamma_{\text{BOT}}$.

Γ_{MID}.

$$\begin{aligned} A \notin \Gamma_{\text{TOP}} &\iff \exists A^+ \notin \Gamma, A \subseteq A^+, \textit{ and } |A^+| > 0.54n, \\ A \in \Gamma_{\text{MID}} &\iff (A \in \Gamma \textit{ and } 0.46n \leq |A| \leq 0.54)\,, \textit{ or } |A| > 0.54n, \\ A \in \Gamma_{\text{BOT}} &\iff \exists A^- \in \Gamma, A^- \subseteq A, \textit{ and } |A^-| < 0.46n. \end{aligned}$$

Claim 5.7 (Liu and Vaikuntanathan [123])**.** $\Gamma = \Gamma_{\text{TOP}} \wedge (\Gamma_{\text{MID}} \vee \Gamma_{\text{BOT}})$.

By simple closure properties of secret-sharing schemes, we can realize by realizing $\Gamma_{\text{TOP}}, \Gamma_{\text{MID}}, \Gamma_{\text{BOT}}$.

Lemma 5.8 ([123])**.** *Let Γ be an access structure and assume that Γ_{MID} can be realized by secret-sharing schemes with share size $2^{c'n}$ for some constant $0 < c' < 1$. Then, Γ can be realized by a secret-sharing scheme with share size $2^{\max\{0.996,c'\}\cdot n}$.*

Proof. By Claim 5.7, $\Gamma = \Gamma_{\text{TOP}} \wedge (\Gamma_{\text{MID}} \vee \Gamma_{\text{BOT}})$. The access structure Γ_{BOT} has at most $\binom{n}{\leq 0.46n} \leq 2^{h(0.46)n} \leq 2^{0.996n}$ authorized sets (where $h(p)$ is the binary entropy function) and, by the first secret-sharing scheme of [107] described in Section 4.2, the access structure Γ_{BOT} can be realized by a scheme with share size $2^{0.996n}$. The access structure Γ_{TOP} has at most $\binom{n}{\geq 0.54n} \leq 2^{0.996n}$ unauthorized sets and, by the second secret-sharing scheme of [107] described in Section 4.2, the access structure Γ_{TOP} can be realized by a scheme with share size $2^{0.996n}$.

By standard closure properties of secret-sharing schemes, realizing Γ can be reduced to realizing Γ_{TOP}, Γ_{BOT}, and Γ_{MID}, that is, to share a secret $s \in \{0,1\}$, the dealer chooses a random bit r, shares $r \oplus s$ with a scheme realizing Γ_{TOP} and independently shares r with a scheme realizing Γ_{MID} and with a scheme realizing Γ_{BOT}. We obtain a secret-sharing scheme realizing Γ with share size as claimed. □

5.4.2 Balancing the Sizes of Authorized Sets in the Access Structure Γ_{MID}

To realize the access structure Γ_{MID}, we defined balanced access structures $\Gamma_{\text{MID},B}$ in Definition 5.9 and show how to represent Γ_{MID} as a union of a polynomial number of balanced access structures. In Section 5.4.3, we

show how to realize the access structure $\Gamma_{\mathrm{MID},B}$ using the (t, N)-robust secret-sharing scheme Π_{Robust}. By closure properties of secret-sharing schemes, Γ_{MID} can be realized using the schemes Π_{Robust}, and, hence, the access structure Γ can be realized using the scheme Π_{Robust}.

Definition 5.9 (The Access Structure $\Gamma_{\mathrm{MID},B}$)**.** *Let Γ be an n-party access structure and B be a subset of parties. The access structure $\Gamma_{\mathrm{MID},B}$ is the access structure whose minimal authorized sets are all subsets of parties of size greater than $0.54n$, and all subsets of parties that contain authorized subsets $A' \in \Gamma$ of size between $0.46n$ and $0.54n$ that contain exactly $\lfloor |A'|/2 \rfloor$ of their parties from B. Formally, we define $\Gamma_{\mathrm{MID},B}$ as the following access structure*

$$\Gamma_{\mathrm{MID},B} = \left\{ A : \exists A' \in \Gamma, A' \subseteq A, 0.46n \leq |A'| \leq 0.54n, \text{ and } |A' \cap B| = \lfloor |A'|/2 \rfloor \right\} \cup \left\{ A : |A| > 0.54n \right\}.$$

Example 5.10. Consider the access structure Γ with 4 parties p_1, p_2, p_3, p_4, where

$$\Gamma = \left\{ \left\{ p_1, p_2 \right\}, \left\{ p_1, p_3 \right\} \right\} \cup \left\{ A \subseteq \left\{ p_1, p_2, p_3, p_4 \right\} : |A| \geq 3 \right\}.$$

Then, the sets in $\Gamma_{\mathrm{MID},\{p_1,p_2\}}$ of size 2 are the sets of size 2 in Γ that contain exactly 1 party from $\left\{ p_1, p_2 \right\}$, namely, $\left\{ p_1, p_3 \right\} \in \Gamma_{\mathrm{MID},\{p_1,p_2\}}$ and $\left\{ p_1, p_2 \right\} \notin \Gamma_{\mathrm{MID},\{p_1,p_2\}}$. Similarly, $\left\{ p_1, p_3 \right\} \notin \Gamma_{\mathrm{MID},\{p_2,p_4\}}$ and $\left\{ p_1, p_2 \right\} \in \Gamma_{\mathrm{MID},\{p_2,p_4\}}$. Notice that

$$\Gamma = \Gamma_{\mathrm{MID},\{p_1,p_2\}} \vee \Gamma_{\mathrm{MID},\{p_2,p_4\}}.$$

To realize Γ_{MID} using secret-sharing schemes for $\Gamma_{\mathrm{MID},B}$, we use a family of subsets, in which every set of medium size is equally partitioned by at least one of the subsets in the family. In the next lemma, we assume without loss of generality that the number of parties n is even (this can be done by adding a dummy party).

Lemma 5.11. *Let n be an even integer and Γ be an n-party access structure. There are $w = O(n^{3/2})$ subsets $B_1, \ldots, B_w \subseteq P$, each of them of size $n/2$, such that $\Gamma = \vee_{1 \leq i \leq w} \Gamma_{\mathrm{MID},B_i}$. In particular, if for every $B \subset P$ of size $n/2$, the access structure $\Gamma_{\mathrm{MID},B}$ can be realized by a secret-sharing scheme with share size $2^{(c'+o(1))n}$ for some constant $0 < c' < 1$, then Γ_{MID} can be realized by a secret-sharing scheme with share size $2^{(c'+o(1))n}$.*

Proof. Fix a set A such that $0.46n \leq |A| \leq 0.54n$ and pick at random with a uniform distribution a set B of size $n/2$. Then

$$\Pr\left[|A \cap B| = \left\lfloor \frac{|A|}{2} \right\rfloor \right] = \frac{\binom{|A|}{\lfloor |A|/2 \rfloor} \binom{n-|A|}{n/2-\lfloor |A|/2 \rfloor}}{\binom{n}{n/2}} = \Theta\left(\frac{2^{|A|}/\sqrt{n} \cdot 2^{n-|A|}/\sqrt{n}}{2^n/\sqrt{n}} \right) = \Theta\left(\frac{1}{\sqrt{n}} \right)$$

(this follows from the fact that $\binom{n}{\lfloor n/2 \rfloor} = \Theta(2^n/\sqrt{n})$ and $|A| = \theta(n)$). By a simple probabilistic proof, there exist $w = \Theta(n^{3/2})$ subsets $B_1, \ldots, B_w \subseteq P$, where $|B_i| = n/2$ for every $i \in [w]$, such that for every subset A such that $0.46n \leq |A| \leq 0.54n$, it holds that $|A \cap B_i| = \lfloor |A|/2 \rfloor$ for at least one $i \in [w]$. Together with the observation that $\Gamma_{\mathrm{MID},B} \subseteq \Gamma$, this implies that $\Gamma_{\mathrm{MID}} = \vee_{i=1}^{w} \Gamma_{\mathrm{MID},B_i}$.

For every $i \in [w]$, we independently share the secret s using the secret-sharing scheme realizing the access structure Γ_{MID,B_i}; by the assumption of the lemma, the share size of this scheme is $2^{(c'+o(1))n}$. The combined scheme is a secret-sharing scheme realizing the access structure Γ_{MID} in which the share size is $O(n^{3/2}) \cdot 2^{(c'+o(1))n} = 2^{(c'+o(1))n}$. □

5.4.3 Realizing $\Gamma_{\text{MID},B}$

To complete the description of a scheme realizing Γ, we present a secret-sharing scheme realizing the access structure $\Gamma_{\text{MID},B}$. This scheme is similar to the secret-sharing scheme described in Section 5.1 (i.e., constructing a bipartite graph from $\Gamma_{\text{MID},B}$) with two main differences. First, it uses a more refined graph, containing only edges for minimal authorized sets in $\Gamma_{\text{MID},B}$. Second, it executes two robust secret-sharing schemes for this graph, an (N,t)-robust scheme and a (t,N)-robust scheme. For every unauthorized set T (of size at most $0.54n$), it holds that either $|T \cap B| \leq |T|/2$ or $|T \cap \overline{B}| \leq |T|/2$, which will enable us to take a smaller t.

Assume without loss of generality that n is even. Define

$$U = \left\{ A_1 \subseteq B \,:\, 0.23n \leq |A_1| \leq 0.27n \right\}$$

and

$$V = \left\{ A_2 \subseteq \overline{B} \,:\, 0.23n \leq |A_2| \leq 0.27n \right\}.$$

Let $N = |U| = |V|$. Note that $N = \Theta(2^{n/2})$. Moreover, define the bipartite graph $G = (U, V, E)$, where for vertices $A_1 \in U, A_2 \in V$ there is an edge $(A_1, A_2) \in E$ if and only if $A_1 \cup A_2 \in \Gamma, 0.46n \leq |A_1 \cup A_2| \leq 0.54n$, and $|A_1| = |A_2|$ or $|A_1| = |A_2| - 1$. The scheme $\Pi_{\text{MID},B}$ realizing $\Gamma_{\text{MID},B}$ is described in Figure 5.7.

Lemma 5.12. *Let Γ be an n-party access structure and B be a subset of parties such that $|B| = n/2$. Then, the scheme $\Pi_{\text{MID},B}$ described in Figure 5.7 is a secret-sharing scheme realizing $\Gamma_{\text{MID},B}$ with a one-bit secret in which the share size is $2^{0.993n}$.*

Proof. For the correctness of the scheme, first take a minimal authorized set $A \in \Gamma_{\text{MID},B}$ such that $|A| \leq 0.54n$, that is, $A = A_1 \cup A_2$ for some $A_1 \subseteq B$, $A_2 \subseteq \overline{B}$ such that $A_1 \cup A_2 \in \Gamma$, $0.46n \leq |A_1 \cup A_2| \leq 0.54n$, and $|A_1| = |A_2|$ or $|A_1| = |A_2| - 1$, that is, $(A_1, A_2) \in E$. The parties in $A = A_1 \cup A_2$ can reconstruct the $\mathsf{sh}^1_{A_1}$ and $\mathsf{sh}^1_{A_2}$ from the shares of the (t, N) robust secret-sharing scheme and can reconstruct sh_1 from these shares (since $(A_1, A_2) \in E$). By symmetric arguments, the parties in A can reconstruct sh_2 (using the shares of the (N,t) robust secret-sharing scheme), and, thus, the parties in A can reconstruct the secret s by xoring sh_1 and sh_2. Authorized sets of size greater than $0.54n$ can reconstruct the secret s using the $(0.54n+1)$-out-of-n secret-sharing scheme.

For the security of the scheme, consider an unauthorized set $T \notin \Gamma_{\text{MID},B}$, that is, $T = T_1 \cup T_2$ such that $T_1 \subseteq B, T_2 \subseteq \overline{B}$, and $|T_1 \cup T_2| \leq 0.54n$ (subsets of size greater than $0.54n$ are authorized), and assume without loss of generality that $|T_1| \leq 0.27n$ (otherwise, $|T_2| \leq 0.27n$ and we consider the (N,t)-robust secret-sharing scheme). In the (t, N)-robust secret-sharing scheme, the parties in T_1 know the share of $\mathsf{sh}^1_{T_1'}$

Scheme $\Pi_{\mathrm{MID},B}$

The secret: A bit $s \in \{0,1\}$.

The scheme:

1. Share s among the n parties using a $(0.54n+1)$-out-of-n secret-sharing scheme.
2. Choose a random bit $\mathsf{sh}_1 \in \{0,1\}$ and define $\mathsf{sh}_2 = s \oplus \mathsf{sh}_1$.
3. Let $t = n \cdot 2^{0.164n}$ (this choice of t will be explained later).
4. Construct the above graph $G = (U, V, V)$ for the access structure $\Gamma_{\mathrm{MID},B}$.
5. Share the secret sh_1 using the (t, N) robust secret-sharing scheme Π_{Robust} for G. Let sh^1_C be the share in this scheme of the vertex $C \in U \cup V$.
6. Share the secret sh_2 using the (N, t) robust secret-sharing scheme Π_{Robust} for G. Let sh^2_C be the share in this scheme of the vertex $C \in U \cup V$.
7. For every set $C \in U \cup V$ (where C is a set of parties), independently share sh^1_C and sh^2_C using the $|C|$-out-of-$|C|$ secret-sharing scheme of Example 1.1 among the parties of C.

Figure 5.7: A secret-sharing scheme $\Pi_{\mathrm{MID},B}$ realizing the access structure $\Gamma_{\mathrm{MID},B}$.

for every $T'_1 \in U$ if and only if $T'_1 \subseteq T_1$. That is, they can reconstruct shares of vertices in U (which are sets) for the sets $\mathcal{T} = \left\{T'_1 \in U \,:\, T'_1 \subseteq T_1, |T'_1| \geq 0.23n\right\}$. The number of such subsets is at most

$$t \triangleq \sum_{i=0.23n}^{0.27n} \binom{0.27n}{i} \leq n \cdot \binom{0.27n}{0.23n} \leq n \cdot 2^{h(0.23/0.27)\cdot 0.27n} < n \cdot 2^{0.164n},$$

where $h(\cdot)$ is the binary entropy.

For every $T'_1 \subseteq T_1$ and $T'_2 \subseteq T_2$, we have that $(T'_1, T'_2) \notin E$. Thus, the parties in $T = T_1 \cup T_2$ (which learn the shares on the vertices of $\mathcal{T} \subset U$ and possibly many shares of vertices in V) only learn the shares of the independent set $\mathcal{T} \cup \left\{T'_2 \in V \,:\, T'_2 \subseteq T_2\right\}$ in the (t, N)-robust secret-sharing scheme. Thus, by the (t, N)-robustness of the secret-sharing scheme, the parties in A cannot learn any information on sh_1, and, hence, they cannot learn any information on the secret s.

Overall, in the scheme $\Pi_{\mathrm{MID},B}$, each party p_i gets a share of size $\log(n)$ from the threshold scheme of step 1 and less than $2|U| = 2|V| = O(2^{n/2})$ shares from the secret-sharing scheme of Example 1.1 (two shares for each set C such that $p_i \in C$). Thus, since $t = n \cdot 2^{0.164n} > 2^{n/8} > |U|^{1/4} = |V|^{1/4}$, the share size of the (t, N)-robust secret-sharing scheme is $O(t^3 \log(N)) < 2^{0.493n}$, and the share size of each party in the scheme $\Pi_{\mathrm{MID},B}$ is

$$O(2^{n/2} \cdot t^3 n) < 2^{0.993n}.$$

□

5.5 Putting Everything Together

By Lemmas 5.11 and 5.12, the access structure Γ_{MID} can be realized by a secret-sharing scheme with share size $2^{0.993n}$. By Lemma 5.8 every n-party access structure can be realized by a secret-sharing scheme with share size $2^{(0.996+o(1))n}$. To summarize the construction of the secret-sharing scheme for Γ, we have shown in Lemmas 5.8 and 5.11 that

$$\Gamma = \Gamma_{\mathrm{TOP}} \wedge \left(\bigvee_{1 \leq i \leq w} (\Gamma_{\mathrm{MID},B_i}) \vee \Gamma_{\mathrm{BOT}} \right)$$

(see Figure 5.5). We have proved that $\Gamma_{\mathrm{BOT}}, \Gamma_{\mathrm{TOP}}, \Gamma_{\mathrm{MID},B_1}, \ldots, \Gamma_{\mathrm{MID},B_w}$ can be realized by secret-sharing schemes with share size $2^{(0.996+o(1))n}$. Using the construction of [36] (see Section 4.3) for this formula, we obtain a secret-sharing for Γ.

The construction we described in this monograph is not the most efficient known scheme. The best known construction is summarized below.

Theorem 5.13 (Applebaum and Nir [9])**.** *Every n-party access structure can be realized by a secret-sharing scheme with share size* $1.5^{(1+o(1))n} < 2^{0.585n}$.

The first ingredient to achieve the improved scheme is using robust secret-sharing schemes for $\sqrt{N}$-partite $\sqrt{N}$-hypergraphs (i.e., each minimal authorized set is a hyperedge of size $\sqrt{N}$ containing exactly

one vertex from each part). To construct such schemes, we start with the non-robust secret-sharing schemes for such hypergraphs from [125] (in [125] they are described as k-server conditional disclosure of secrets (CDS) protocols; these two notions are equivalent). We then "immunize" this scheme and make it robust using a generalization of the ideas described in Section 5.3. Finally, a better decomposition is used (instead of Claim 5.7).

Chapter 6

Secret Sharing and Secure Multi-Party Computation

Secret-sharing schemes are a basic building block in the construction of many cryptographic protocols. In this chapter we demonstrate the use of secret-sharing schemes to construct secure multi-party computation (MPC) protocols for general functions. The purpose of this chapter is to give some ideas on how to use secret-sharing schemes to construct secure multi-party computation protocols; we will be informal in our discussion, definitions, and proofs. For simplicity, we concentrate on the case that the "bad" parties are semi-honest, that is, the parties follow the instructions of the protocol; however, at the end of the protocol some of them might collude and try to deduce information from the messages they got. The protocols that we describe are secure against an all-powerful adversary, that is, they guarantee information-theoretic security. MPC protocols were introduced by [185, 97] in the computational setting. MPC protocols with information-theoretic security (as we consider in this monograph) were constructed in [34, 57, 153]. The reader can find more information on MPC protocols in the computational setting in, e.g., [122]. The book of Cramer et al. [65] discusses information-theoretic MPC protocols and their constructions via secret-sharing schemes.

Definition 6.1 (Secure Computation in the Semi-Honest Model (Informal))**.** *Let S be a finite domain of inputs. There are n parties $p_1, \ldots, p_n$; each party p_j holds a private input $x_j \in S$. At most t of the parties are semi-honest, where $t < n$; we assume that all other parties are honest. The parties want to compute some function $f(x_1, \ldots, x_n)$ by exchanging messages on private channels according to some protocol $\mathcal{P}$. A protocol is t-private if it satisfies the following two requirements.*

Correctness. *At the end of the protocol each party outputs $f(x_1, \ldots, x_n)$.*

Security (informal). *Every coalition T of at most t parties cannot learn any information not implied by the inputs $\langle x_j \rangle_{p_j \in T}$ and the output of the function. This property is formalized by the existence of a simulator that, given the inputs and outputs of the parties in T, generates the view in the protocol of the parties in T (without seeing the inputs and the randomness of the parties not in T).*

In the rest of this chapter we describe a private protocol for computing general functions. First, as a warm-up, we describe a private protocol for modular addition. Next, we discuss homomorphic properties of Shamir's secret-sharing scheme. We show that these properties enable the parties to compute without any interaction shares of the sum of two shared secrets. Then, we show a protocol that privately computes shares of the product of two shared secrets. Combining these protocols we get an efficient protocol for computing any function that can be computed by a small arithmetic circuit. Such protocols with information-theoretic security were first presented in [34, 57]. The exact protocol we present here is from [94].

6.1 A Private Protocol for Addition

As a warm-up, we describe in Figure 6.1 an n-private protocol for computing the sum of n elements in $\mathbb{F}_q$ for some prime-power q, that is, each party holds an input $x_i \in \mathbb{F}_q$ and the parties want to privately compute $\sum_{i=1}^{n} x_i$.

Protocol ADD

1. Each party p_j shares x_j with the n-out-of-n secret-sharing scheme of Example 1.1 over $\mathbb{F}_q$, that is, p_j chooses $n-1$ random field elements $\mathsf{sh}_1^j, \ldots, \mathsf{sh}_{n-1}^j \in \mathbb{F}_q$ and computes $\mathsf{sh}_n^j \leftarrow x_j - \sum_{i=1}^{n-1} \mathsf{sh}_i^j$. For each party p_i, party p_j sends sh_i^j to p_i.
2. Each party p_i computes $\mathsf{sh}_i \leftarrow \sum_{j=1}^{n} \mathsf{sh}_i^j$ (i.e., sh_i is the sum of the i-th shares of $x_1, \ldots, x_n$) and sends sh_i to all parties.
3. Each party p_j outputs $z \leftarrow \sum_{i=1}^{n} \mathsf{sh}_i$.

Figure 6.1: A protocol for privately computing the sum of n field elements. Addition and subtraction are in $\mathbb{F}_q$.

Claim 6.2. *Protocol ADD described in Figure 6.1 is an n-private protocol for addition over the field $\mathbb{F}_q$.*

Proof. The correctness of the protocol follows from the additivity of the n-out-of-n secret-sharing scheme, that is, if each party sums the shares of the n inputs, the result is an n-out-of-n secret-sharing of the sum of the n inputs. Formally,

$$\sum_{i=1}^{n} \mathsf{sh}_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \mathsf{sh}_i^j \right) = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} \mathsf{sh}_i^j \right) = \sum_{j=1}^{n} x_j .$$

The proof of security is not complicated but rather technical. Notice that $n-1$ parties holding their inputs and getting the sum z of all inputs can compute the input of the remaining party, thus in this case there are no requirements. We need to show that for every $1 \leq t \leq n-2$, a set of t parties, say $p_1, \ldots, p_t$, does not learn information not implies by its inputs $x_1, \ldots, x_t$ and the sum z, that is it cannot distinguish between inputs

$x_{t+1}, \ldots, x_n$ and inputs $x'_{t+1}, \ldots, x'_n$ such that

$$z = \sum_{i=1}^{n} x_i = \sum_{i=1}^{t} x_i + \sum_{i=t+1}^{n} x'_i$$

i.e.,

$$\sum_{i=t+1}^{n} (x_i - x'_i) = 0. \tag{6.1}$$

Consider an execution of Protocol ADD on $x_1, \ldots, x_t, x_{t+1}, \ldots, x_n$, where p_i, for $1 \leq i \leq n$, produced shares such that

$$x_i = \sum_{j=1}^{n} \mathsf{sh}_j^i.$$

In this case the view of $p_1, \ldots, p_t$ is

$$x_1, \ldots, x_t, \left\langle \mathsf{sh}_i^j \right\rangle_{1 \leq i \leq n, 1 \leq j \leq t}, \left\langle \mathsf{sh}_i^j \right\rangle_{1 \leq i \leq t, t+1 \leq j \leq n}, \left\langle \mathsf{sh}_i = \sum\nolimits_{j=1}^{n} \mathsf{sh}_i^j \right\rangle_{1 \leq i \leq n}.$$

Now consider an execution of Protocol ADD on $x_1, \ldots, x_t, x'_{t+1}, \ldots, x'_n$, where each p_j produced the following shares:

$$\mathsf{sh'}_i^j = \begin{cases} \mathsf{sh}_i^j & \text{If } 1 \leq i \leq n-1 \text{ or } 1 \leq j \leq t \\ \mathsf{sh}_n^j + x'_j - x_j & \text{If } i = n \text{ and } t+1 \leq j \leq n. \end{cases}$$

That is, an honest party p_j only changes the share that it sends to the honest party p_n and all shares sh_i^j sent to the semi-honest parties $p_1, \ldots, p_t$ do not change. Note that in the second execution p_j, for $t+1 \leq i \leq n$, produces shares of x'_j as $\sum_{i=1}^{n-1} \mathsf{sh}_i^j + \mathsf{sh'}_n^j = x'_i$. Furthermore, in the second execution, party p_i, for $t+1 \leq i \leq n-1$, sends to $p_1, \ldots, p_t$ the same shares sh_i as in the first execution. Finally, in the second execution p_n sends to $p_1, \ldots, p_t$ the same share sh_n as in the first execution since

$$\mathsf{sh'}_n = \sum_{j=1}^{t} \mathsf{sh}_n^j + \sum_{j=t+1}^{n} \mathsf{sh'}_n^j = \sum_{j=1}^{n} \mathsf{sh}_n^j + \sum_{j=t+1}^{n} (x'_j - x_j) = \mathsf{sh}_n + \sum_{j=t+1}^{n} (x'_j - x_j) = \mathsf{sh}_n$$

(where the last equality follows from (6.1). Thus, $p_1, \ldots, p_t$ cannot distinguish between the inputs $x_{t+1}, \ldots, x_n$ and $x'_{t+1}, \ldots, x'_n$ and privacy follows. □

6.2 Homomorphic Properties of Shamir's Secret-Sharing Scheme

We next prove that Shamir's scheme is additive. This is a special case of additivity of linear secret-sharing schemes discussed in Lemma 4.13. Furthermore, we show that this scheme also has a weaker multiplicative property.

Claim 6.3. *Let t, n be integers such that $t < n$, $\mathbb{F}_q$ be a finite field with more than n elements, and $s_1, s_2 \in \mathbb{F}_q$ be two secrets. For $i \in \{1,2\}$, let $\mathsf{sh}_{i,1}, \ldots, \mathsf{sh}_{i,n}$ be a sharing of s_i using Shamir's $(t+1)$-out-of-n scheme (see Section 2.2). Then, $\mathsf{sh}_{1,1} + \mathsf{sh}_{2,1}, \ldots, \mathsf{sh}_{1,n} + \mathsf{sh}_{2,n}$ are shares of the secret $s_1 + s_2$ in Shamir's $(t+1)$-out-of-n scheme. In addition, if $t < n/2$, then $\mathsf{sh}_{1,1} \cdot \mathsf{sh}_{2,1}, \ldots, \mathsf{sh}_{1,n} \cdot \mathsf{sh}_{2,n}$ are shares of the secret $s_1 \cdot s_2$ in Shamir's $(2t+1)$-out-of-n scheme.*

Proof. Let Q_1 and Q_2 be the polynomial of degree at most t generating the shares $\mathsf{sh}_{1,1}, \ldots, \mathsf{sh}_{1,n}$ and $\mathsf{sh}_{2,1}, \ldots, \mathsf{sh}_{2,n}$ respectively, that is $Q_i(0) = s_i$ and $Q_i(\alpha_j) = \mathsf{sh}_{i,j}$ for $i \in \{1,2\}$ and $1 \leq j \leq n$ (where $\alpha_1, \ldots, \alpha_n$ are defined in Section 2.2). Define $Q(x) = Q_1(x) + Q_2(x)$. This is a polynomial of degree at most t such that $Q(0) = Q_1(0) + Q_2(0) = s_1 + s_2$ and $Q(\alpha_j) = \mathsf{sh}_{1,j} + \mathsf{sh}_{2,j}$, that is, this is a polynomial generating the shares $\mathsf{sh}_{1,1} + \mathsf{sh}_{2,1}, \ldots, \mathsf{sh}_{1,n} + \mathsf{sh}_{2,n}$ given the secret $s_1 + s_2$.

Similarly, let $R(x) = Q_1(x) \cdot Q_2(x)$. When $t < n/2$, this is a polynomial of degree at most $2t < n$ generating the shares $\mathsf{sh}_{1,1} \cdot \mathsf{sh}_{2,1}, \ldots, \mathsf{sh}_{1,n} \cdot \mathsf{sh}_{2,n}$ given the secret $s_1 \cdot s_2$.[18] □

6.3 Computing the Sharing of the Sum of Two Shared Secrets

Assume that two secrets x_1 and x_2 are shared using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Using Claim 6.3, each party can compute a share of the sum of the secrets without any communication, as described in Figure 6.2.

Protocol Sum

Input of party p_j: Shares $\mathsf{sh}_{1,j}$ and $\mathsf{sh}_{2,j}$ of the secrets x_1 and x_2 respectively.

Computation step: Each party p_j computes $\mathsf{sh}_j = \mathsf{sh}_{1,j} + \mathsf{sh}_{2,j}$.

Figure 6.2: A protocol for computing shares of the sum of two shared secrets.

6.4 Computing the Product of Two Shared Secrets

Assume that two secrets x_1 and x_2 are shared using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Using Claim 6.3, the parties can compute shares of the product $x_1 \cdot x_2$ in a $(2t+1)$-out-of-n secret-sharing scheme. In Figure 6.3, we show that, by using one round of interaction, the parties can compute shares of the product $x_1 \cdot x_2$ in Shamir's $(t+1)$-out-of-n secret-sharing scheme (without learning the product itself). In this case, we assume that there are t semi-honest parties, where $n = 2t+1$ (that is, there is a majority of honest parties).

[18] While $Q(x)$ is a uniformly distributed polynomial such that $Q(0) = s_1 + s_2$, the polynomial $R(x)$ is *not* uniformly distributed (that is, $R(x)$ is product of two polynomials of degree t). For the protocols we present, this does not cause any problems.

Protocol Product

Input of party p_j. Shares $\mathsf{sh}_{1,j}$ and $\mathsf{sh}_{2,j}$ of the secrets x_1 and x_2 respectively in Shamir's $(t+1)$-out-of-n secret-sharing scheme.

Output. Shares $u_1, \ldots, u_m$ in a t-out-of-n secret-sharing scheme of the secret $x_1 \cdot x_2$.

Step I. Each party p_j computes $\mathsf{sh}_j = \mathsf{sh}_{1,j} \cdot \mathsf{sh}_{2,j}$ and shares sh_j using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Denote the resulting shares by $q_{j,1}, \ldots, q_{j,n}$. Party p_j sends $q_{j,\ell}$ to p_ℓ.

Step II. Let $\beta_1, \ldots, \beta_n$ be the constants defined in (2.3) for the reconstruction of the secret in Shamir's $(2t+1)$-out-of-n scheme. Each party p_ℓ computes $u_\ell = \sum_{j=1}^{n} \beta_j q_{j,\ell}$.

Figure 6.3: A protocol for computing shares of the product of two shared secrets.

Lemma 6.4. *Let t, n be two integers such that $n = 2t+1$. Protocol Product described in Figure 6.3 is a t-private protocol whose input are shares in Shamir's $(t+1)$-out-of-n secret-sharing scheme of two secrets x_1, x_2 and output is random shares in the same scheme of the secret $x_1 \cdot x_2$.*

Proof. We first explain why this protocol is correct. By Claim 6.3, $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$ are shares of $x_1 \cdot x_2$ in a Shamir's $(2t+1)$-out-of-n scheme. By (2.3), since $n = 2t+1$, the constants $\beta_1, \ldots, \beta_n$ for the reconstruction in Shamir's $(2t+1)$-out-of-n secret-sharing scheme exist and

$$x_1 \cdot x_2 = \sum_{j=1}^{n} (\beta_j \cdot \mathsf{sh}_j).$$

As $q_{j,1}, \ldots, q_{j,n}$ are shares in Shamir's $(t+1)$-out-of-n scheme of the secret sh_j, Claim 6.3 implies that $u_1, \ldots, u_\ell$ are shares of the linear combination $\sum_{j=1}^{n} (\beta_j \cdot \mathsf{sh}_j) = x_1 \cdot x_2$. Furthermore, since (for example) party p_1 chooses random shares of sh_1, the parties hold random shares of $x_1 \cdot x_2$.

Informally, the security of the protocol follows from the fact that any coalition of size at most t only sees shares in Shamir's $(t+1)$-out-of-n secret-sharing scheme, i.e., it does not gain any information on the inputs of parties not in T. We next provide a formal proof. For simplicity of the notation, we consider the set $T = \{1, \ldots, t\}$, assume that Shamir's scheme is over the field $\mathbb{F}_p$, where $p > n$ is a prime, and use $\alpha_i = i$ in Shamir's scheme. The view of T is

$$\left\langle \mathsf{sh}_{1,j}, \mathsf{sh}_{2,j} \right\rangle_{1 \le j \le t}, \left\langle q_{j,\ell} \right\rangle_{1 \le j \le t, 1 \le \ell \le n}, \text{ and } \left\langle q_{j,\ell} \right\rangle_{t+1 \le j \le n, 1 \le \ell \le t}.$$

We will show that this view could have been generated for every pair of secrets x_1, x_2. First consider the two polynomials Q_{x_1}, Q_{x_2} of degree at most t such that for every $b \in \{1, 2\}$

$$\forall_{1 \le j \le t}\ Q_{x_b}(j) = \mathsf{sh}_{b,j} \text{ and } Q_{x_b}(0) = x_b,$$

that is, Q_{x_b} generates the shares $\left\langle \mathsf{sh}_{b,j} \right\rangle_{1 \le j \le t}$ for the secret x_b. Let $R(x) = Q_{x_1}(x) \cdot Q_{x_2}(x)$, i.e., $R(x)$ generates shares in a $(2t+1)$-out-of-n secret-sharing scheme for the secret $x_1 \cdot x_2$. Furthermore, let $Q_1, \ldots, Q_n$ be the polynomials of degree at most t such that for every $1 \le i \le n$

$$\forall_{1 \le j \le t}\ Q_i(j) = q_{i,j} \text{ and } Q_i(0) = R(i).$$

By Claim 2.3 all the above polynomials exist and they are unique. Furthermore, they generate the view of T for the secrets x_1, x_2. Thus, the view of T is generated with the same probability for every pair of secrets x_1, x_2. □

6.5 Privately Computing an Arithmetic Circuit

Using the above protocols, we show how to securely compute any function represented by an arithmetic circuit assuming that $n = 2t + 1$. Recall that any function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ can be represented by an arithmetic circuit over $\mathbb{F}_q$ (with addition and multiplication gates and fan-in 2). For a definition of arithmetic circuits, the reader is referred to Appendix A.1.

In Figure 6.4, we describe a secure protocol for evaluating the function computed by an arithmetic circuit, where each party p_j holds an input $x_j \in \mathbb{F}_q$. The number of rounds in this protocol is linear in the number of nodes.[19] More formally, let $G_1, G_2, \ldots, G_\ell$ be the nodes of a circuit sorted according to some topological order (that is, if there exists an edge from G_j to G_i, then $j < i$). Assume that, for $1 \le i \le n$, the node G_i is labeled by the variable x_i (i.e., it is an input node). The protocol for computing the arithmetic circuit keeps intermediate values as shares of a $(t+1)$-secret-sharing scheme. In the beginning of the protocol, each party shares its input. Thereafter, the protocol proceeds in rounds, where in the beginning of round i the parties hold shares of a $(t+1)$-out-of-n secret-sharing scheme of the two inputs of the node G_i, and in the end of round i the parties hold shares of a $(t+1)$-out-of-n secret-sharing scheme of the output of the node G_i; for an addition gate this is done by local computation (using Claim 6.3) and for a multiplication gate this is done using Protocol Product. At the end of the protocol, the output is reconstructed from the shares.

By the correctness of the addition and multiplication protocols, at the end of round i, the parties hold shares of the output of the node G_i. Thus, at the end of the protocol they hold shares of the output of the circuit, and s is the correct value for the output of the protocol. On the other hand, by Lemma 6.4, in each stage any coalition of at most t parties sees at most t shares of a $(t+1)$-out-of-n secret-sharing scheme and at the end of the protocol they see random shares of the output of the circuit, thus, very informally, the set does not learn information not implied by the inputs of the set and the output of the circuit.

6.6 Extensions to Other Models

The protocol we described above assumes that the corrupted parties are semi-honest. A more realistic assumption is that the parties can deviate from the protocol and send any messages that might help them. Such

[19] The number of rounds can be reduced to the depth of the circuit, i.e., the longest path from a leaf to the root.

Protocol MPC

Input of party p_j. An element $x_j \in \mathbb{F}_q$, where $\mathbb{F}_q$ is a finite field such that $|\mathbb{F}| > n$.

Initialization. Each party p_j shares x_j using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Denote the resulting shares by $\mathsf{sh}_1^j, \ldots, \mathsf{sh}_n^j$. Party p_j sends sh_i^j to p_i.

Computation stages. For $m = n + 1$ to ℓ compute shares of the output of the node G_m as follows:

- Assume that the incoming edges into node G_m are from nodes G_{m_1} and G_{m_2}, where $m_1, m_2 < m$ and the parties holds shares $\mathsf{sh}_1^{m_1}, \ldots, \mathsf{sh}_n^{m_1}$ and $\mathsf{sh}_1^{m_2}, \ldots, \mathsf{sh}_n^{m_2}$ of the outputs of these nodes.
- If G_m is an addition gate, each party p_i locally computes $\mathsf{sh}_i^m = \mathsf{sh}_i^{m_1} + \mathsf{sh}_i^{m_2}$ as the share of the output of the node G_m.
- If G_m is a multiplication gate, the parties use the one-round protocol described in Section 6.4 to compute shares of the product of the outputs of nodes G_{m_1} and G_{m_2}.

Reconstruction. Each party p_i, for $1 \leq i \leq t + 1$, sends its share sh_i^ℓ to all parties. Each party p_j reconstructs the output from the shares $\mathsf{sh}_1^\ell, \ldots, \mathsf{sh}_{t+1}^\ell$ using the reconstruction procedure of Shamir's $(t + 1)$-out-of-n secret-sharing scheme.

Figure 6.4: A secure multi-party computation (MPC) protocol for computing a function represented by an arithmetic circuit.

parties are called malicious. For example, in the multiplication protocol, a party that should share sh_j can send shares that are not consistent with any secret. Furthermore, in the reconstruction step in the arithmetic circuit protocol, a party can send a "wrong" share. To cope with malicious behavior, the notion of *verifiable secret sharing* was introduced by Chor et al. [61]. Such information-theoretic schemes were constructed, see [94] for a partial list of such constructions. We will not elaborate on verifiable secret sharing in this monograph.

In the definition of secure computation, we assumed that there is a parameter t, and an adversary can control any coalition of size at most t. This assumes that all parties are as likely to be corrupted. Hirt and Maurer [106] considered a more general scenario in which there is an access structure, and the adversary can control any set of parties not in the access structure. That is, they required that any set not in the access structure cannot learn information not implied by the inputs of the parties in the set and the output of the function. Similarly to the requirement that $2t < n$ in the protocol we described above, secure computation against semi-honest parties is possible for general functions iff the union of every two sets not in the access structure does not cover the entire set of parties [106]; such access structure is called Q^2. For every Q^2 access structure Γ, Cramer et al. [64] showed that using any linear secret-sharing scheme realizing Γ, one can construct a protocol for computing any arithmetic circuit such that any set not in the access structure cannot learn any information; the complexity of the protocol is linear in the size of the circuit. Their protocol is similar to the protocol we described above, where for addition gates every party does local computation. Multiplication gates are also dealt in a similar way as in the threshold case in Protocol MPC; however, the choice of the constants $\beta_1, \ldots, \beta_n$ is more involved. The protocol of Cramer et al. [64] demonstrates the need for secret-sharing schemes for general access structures.

Chapter 7

Lower Bounds on the Size of the Shares

The best known constructions of information-theoretic secret-sharing schemes for general n-party access structures (e.g., [123, 6, 8, 9]) have share size $2^{O(n)}$. It is not known if this is the best possible. Lower bounds for secret-sharing schemes have been proved in, e.g., [112, 53, 44, 77, 66, 67, 43]. However, these lower bounds are far from the exponential upper bounds. The best lower bound was proved by Csirmaz [66, 67], who proved that for every n there exists an n-party access structure such that every secret-sharing scheme realizing it has total information ratio $\Omega(n^2/\log(n))$. In Sections 7.2 to 7.4, we review this proof. For linear secret-sharing schemes, the situation is much better – for every n there exist explicit n-party access structures such that every linear secret-sharing scheme realizing them has exponential information ratio, i.e., for every size of secrets ℓ the size of the shares is $\ell \cdot 2^{\Omega(n)}$ [158, 149, 150] (improving on previous lower bounds of [23, 11, 89, 90]). Furthermore, for almost all n-party access structures, the size of the shares in every linear secret-sharing scheme realizing them is at least $2^{0.5n}$ [11]. To demonstrate some of the ideas of these proofs, we present in Section 7.6 a lower bound proof of $n^{\Omega(\log(n))}$ from [90], and in Section 7.5 the lower bound for almost all access structures.

7.1 A Simple Lower Bound

Karnin et al. [112] have showed that for each non-redundant party p_j (that is, a party that appears in at least one minimal authorized set) $H(S_j) \geq H(S)$; we prove this result in Lemma 7.2. Karnin et al.'s result implies that the size of the share of the party is at least the size of the secret. We next give a direct proof of the latter result. We believe that the combinatorial proof we present for Lemma 7.1 is more intuitive than the entropy-based proof of Lemma 7.2 (at least for readers not familiar with information theory).

Lemma 7.1. *Let p_j be a non-redundant party in Γ (i.e., there exists an authorized set $B \in \Gamma$ such that $B \setminus \{p_j\} \notin \Gamma$) and let Π be any secret-sharing scheme realizing Γ, where S and S_j are the domains of secrets and of the shares of p_j respectively. Then, $|S_j| \geq |S|$.*

Proof. Let B be a minimal authorized set in Γ containing p_j, that is $B \in \Gamma$ and $B' \stackrel{\text{def}}{=} B \setminus \{p_j\} \notin \Gamma$. Assume

that $|S_j| < |S|$. Fix any vector of shares $\langle \mathsf{sh}_i \rangle_{p_i \in B'}$ for the parties of B' that has positive probability (given some secret $s_0 \in S$). By the security property, this vector of shares should have positive probability given any secret $s \in S$. That is, for every $s \in S$, there is a share $\mathsf{sh}_j^s \in S_j$ such that $\langle \mathsf{sh}_i \rangle_{p_i \in B'}$ together with sh_j^s have positive probability given the secret s. Since $|S_j| < |S|$, there are secrets $s_1, s_2 \in S$ such that $s_1 \neq s_2$ and $\mathsf{sh}_j^{s_1} = \mathsf{sh}_j^{s_2}$. Thus, the authorized set B holding the shares $\langle \mathsf{sh}_i \rangle_{p_i \in B'}$ and $\mathsf{sh}_j^{s_1}$ errs in the reconstruction for at least one of the secrets s_1 and s_2, contradicting the correctness of the scheme. □

7.2 Lower Bounds Using the Entropy

Starting from the works of Karnin et al. [112] and Capocelli et al. [53], the entropy was used to prove lower bounds on the share size in secret-sharing schemes, e.g., [44, 77, 66, 67]. To prove lower bounds on the information ratio of secret-sharing schemes, they use Definition 3.7 – the alternative definition of secret sharing via the entropy function – and use properties of the entropy function as well as the correctness and security of secret-sharing schemes. For a background on the entropy and its properties, the reader can consult Appendix A.3 and any book on information theory, e.g., [63]. Recall that, given a secret-sharing scheme Π and some distribution on its secrets, $\mathcal{S}$ is the random variable containing the secret and $\mathcal{S}_A$ is a random variable containing the shares of the set A. To simplify notations, in the sequel we denote $H(\mathcal{S}_A)$ by $H(A)$ for any set of parties $A \subseteq \{p_1, \ldots, p_n\}$. Furthermore, we denote $H(\mathcal{S}_A \mathcal{S})$ by $H(A\mathcal{S})$.

We next prove a stronger version of Lemma 7.1, stating that the entropy of the share of p_j is at least $\log(|S_j|)$.

Lemma 7.2. *Let p_j be a non-redundant party in Γ, let Π be any secret-sharing scheme realizing Γ with domain of secrets S. For any distribution on the secrets $\mathcal{S}$, let $\mathcal{S}_j$ be the random variable representing the share of p_j. Then, $H(\mathcal{S}_j) \geq \log(|S|) \geq H(\mathcal{S})$.*

Proof. Since p_j is non-redundant, by Definition 3.4, the probability distribution of the share of p_j is independent of the distribution on the secrets; we thus can assume that the secret is uniformly distributed. Let B be a minimal authorized set in Γ containing p_j; in particular, $B \in \Gamma$ and $B' \stackrel{\text{def}}{=} B \setminus \{p_j\} \notin \Gamma$. On one hand, by the definition of the conditional entropy (A.3), the security of Π (3.6), and the correctness of Π (3.5),

$$H(\mathcal{S}|B') + H(\{p_j\}|B') - H(\mathcal{S}, \{p_j\}|B') = H(\mathcal{S}|B') - H(\mathcal{S}|\{p_j\}, B') = H(\mathcal{S}) - 0 = H(\mathcal{S}).$$

On the other hand, by the definition of the conditional entropy (A.3) and by (A.4),

$$H(\mathcal{S}|B') + H(\{p_j\}|B') - H(\mathcal{S}, \{p_j\}|B') = H(\{p_j\}|B') - H(\{p_j\}|\mathcal{S}, B') \leq H(\{p_j\}|B') \leq H(\{p_j\}).$$

Thus, $H(\mathcal{S}_j) = H(\{p_j\}) \geq H(\mathcal{S}) = \log(|S|)$ (the last equality follows from the fact that the secret is uniformly distributed in S). □

Fujishige [88] has observed that if we take n jointly distributed random variables, the entropy of subsets of these variables is a polymatroid, i.e., it is non-negative, monotone, and submodular. The next theorem

explains how to use these properties of the entropy and the correctness and security of a secret-sharing scheme to prove lower bounds on the share size. In the lower bounds proof, we assume a uniform distribution on the secrets, that is, $H(S) = \log(|S|)$. As proved in Claim 3.8, this assumption is without loss of generality. As the entropy is bounded by the log of the support (A.2), for every j, $H(\{p_j\}) \leq \log(|S_j|)$, thus, the information ratio of the scheme, that is, $\max_{1\leq j\leq n} \log(|S_j|)/\log(|S|)$ is at least $\max_{1\leq j\leq n} H(\{p_j\})/H(S)$.

Theorem 7.3. *Let $A, B \subseteq \{p_1, \ldots, p_n\}$ and Π be a secret-sharing scheme realizing an access structure Γ. The following 4 properties hold:*

Monotonicity.	*If $A \subset B$, then $H(B) \geq H(A) \geq H(\emptyset) = 0$.*
Submodularity.	$H(A) + H(B) \geq H(A \cup B) + H(A \cap B)$.
Strong Monotonicity.	*If $A \notin \Gamma$, $B \in \Gamma$, and $A \subset B$, then* $H(B) \geq H(A) + H(S)$.
Strong Submodularity.	*If $A, B \in \Gamma$ and $A \cap B \notin \Gamma$, then* $H(A) + H(B) \geq H(A \cup B) + H(A \cap B) + H(S)$.

Proof. The monotonicity and submodularity are true for any random variables. The monotonicity follows from (A.6) i.e.,

$$H(B) = H(A, B \setminus A) \geq H(A).$$

The submodularity follows from the definition of conditional entropy (A.3) and the properties of conditional mutual information (A.4):

$$H(A \cup B) - H(A) = H(B|A) \leq H(B|A \cap B) = H(B) - H(A \cap B).$$

For the strong monotonicity observe that by the definition of conditional entropy (A.3), the correctness (3.5), monotonicity, and security (3.6),

$$H(B) = H(BS) - H(S|B) = H(BS) \geq H(AS) = H(S|A) + H(S) = H(A) + H(S).$$

For the strong submodularity, note that if $A, B \in \Gamma$ and $A \cap B \notin \Gamma$, then $H(AS) = H(A)$, $H(BS) = H(B)$, $H((A \cup B)S) = H(A \cup B)$, and $H((A \cap B)S) = H(A \cap B) + H(S)$. Thus, by the submodularity.

$$H(A) + H(B) = H(AS) + H(BS) \geq H((A \cup B)S) + H((A \cap B)S) = H(A \cup B) + H(A \cap B) + H(S).$$

□

To give an example of using Theorem 7.3, we present the lower bound of [53] for the access structure $\Gamma_{\sqcap}$ (defined in Example 3.2).

Theorem 7.4 ([53])**.** *The information ratio of every secret-sharing scheme realizing $\Gamma_{\sqcap}$ is at least* 1.5.

Proof. Let Π be any secret-sharing scheme realizing $\Gamma_{\sqcap}$. By Theorem 7.3,

$$\begin{aligned}
H(\{p_1,p_2\})+H(\{p_2,p_3\}) &\geq H(\{p_1,p_2,p_3\})+H(\{p_2\})+H(S) && \text{strong submodularity,}\\
H(\{p_1,p_3,p_4\}) &\geq H(\{p_1,p_4\})+H(S) && \text{strong monotonicity,}\\
H(\{p_1,p_2,p_3\}) &\geq H(\{p_1,p_3\})+H(S) && \text{strong monotonicity,}\\
H(\{p_1,p_3\})+H(\{p_1,p_4\}) &\geq H(\{p_1,p_3,p_4\})+H(\{p_1\}) && \text{submodularity,}\\
H(\{p_1\})+H(\{p_2\}) &\geq H(\{p_1,p_2\}) && \text{submodularity.}\\
H(\{p_2\})+H(\{p_3\}) &\geq H(\{p_2,p_3\}) && \text{submodularity.}
\end{aligned}$$

Summing all these inequalities, we get $H(\{p_2\})+H(\{p_3\}) \geq 3H(S)$, and the information ratio of the scheme is at least

$$\max\left\{H(\{p_2\}),H(\{p_3\})\right\}/H(S) \geq 1.5.$$

□

7.3 Csirmaz's Lower Bound

We next present Csirmaz's lower bound on the information ratio. The proof has two steps; we first define an access structure whose information ratio is $\Omega(n/\log(n))$. We then define an access structure whose total information ratio is $\Omega(n^2/\log(n))$. The construction in the first step is a generalization of Csirmaz's construction due to Bludo et al. [43].

Definition 7.5 (An Independent Sequence [43])**.** *Let $\ell, n \in \mathbb{N}$ be integers, and let $B = \{p_1,\ldots,p_\ell\}$, $A \subseteq \{p_{\ell+1},\ldots,p_n\}$ be two sets, and Γ be an access structure whose parties are $\{p_1,\ldots,p_n\}$. An independent sequence of length ℓ of Γ is a sequence $A_1,\ldots,A_\ell \subseteq A$ of subsets of A such that:*

- $\{p_1,\ldots,p_i\} \cup A_i \in \Gamma$ *for every* $1 \leq i \leq \ell$.
- $\{p_1,\ldots,p_{i-1}\} \cup A_i \notin \Gamma$ *for every* $1 \leq i \leq \ell$.

Theorem 7.6 ([66, 43])**.** *If an access structure Γ has an independent set of length ℓ, then in any secret-sharing scheme realizing Γ*

$$\sum_{p_j \in A} H(\{p_j\}) \geq (\ell-1)\cdot H(S).$$

In particular, the information ratio of every secret-sharing scheme realizing Γ is at least $\frac{\ell-1}{|A|}$.

Proof. Fix any secret-sharing scheme realizing Γ. Let $A_1,\ldots,A_\ell \subseteq A$ be an independent sequence of length ℓ of Γ and define $B_i = \{p_1\ldots,p_i\}$. Fix an index $1 \leq i \leq \ell-1$ and recall that $B_i \cup A_i \subseteq B_i \cup A \in \Gamma^n$, $B_{i+1} \cup A_{i+1} \in \Gamma^n$, and $B_i \cup A_{i+1} \notin \Gamma^n$. Thus, by the strong submodularity,

$$H(B_i \cup A)+H(B_{i+1} \cup A_{i+1}) \geq H(B_{i+1} \cup A)+H(B_i \cup A_{i+1})+H(S).$$

Furthermore, by submodularity,

$$H(B_i \cup A_{i+1}) + H(B_{i+1}) \geq H(B_{i+1} \cup A_{i+1}) + H(B_i).$$

Summing the last two inequalities, we obtain,

$$H(B_i \cup A) - H(B_i) \geq H(B_{i+1} \cup A) - H(B_{i+1}) + H(S). \tag{7.1}$$

Summing (7.1) for $1 \leq i \leq \ell - 1$ we get that

$$H(B_1 \cup A) - H(B_1) \geq H(B_\ell \cup A) - H(B_\ell) + (\ell - 1)H(S). \tag{7.2}$$

By monotonicity, $H(B_\ell \cup A) - H(B_\ell) \geq 0$. Furthermore, by submodularity, $H(B_1) + H(A) \geq H(B_1 \cup A) + H(\emptyset) = H(B_1 \cup A)$ (since $H(\emptyset) = 0$). Thus,

$$\begin{aligned} H(A) &\geq H(B_1 \cup A) - H(B_1) \\ &\geq H(B_\ell \cup A) - H(B_\ell) + (\ell - 1)H(S) \\ &= (\ell - 1) \cdot H(S). \end{aligned} \tag{7.3}$$

By submodularity, $\sum_{p_j \in A} H(\{p_j\}) \geq H(A)$, thus, there exists at least one party p_j such that $H(\{p_j\}) \geq \frac{\ell-1}{|A|} \cdot H(S)$. This implies that the information ratio of every scheme realizing Γ is at least $\frac{\ell-1}{|A|}$. □

We next define for every $n \in \mathbb{N}$ an n-party access structure whose information ratio is $\Omega(n/\log(n))$. The size of the minimal authorized sets in this access structure is at most $\log(n)$. We define the access structure by specifying its minimal authorized sets.

Definition 7.7 (The Access Structure Γ^n_{Csi})**.** *Fix $n \in \mathbb{N}$ and let k be the largest integer such that $2^k + k \leq n$. Let $B = \{p_1, \ldots, p_{2^k}\}$, $A = \{p_{2^k+1}, \ldots, p_{2^k+k}\}$ (that is, $|A| = k = \Theta(\log(n))$), and $A_1, A_1, \ldots, A_{2^k}$ be all the subsets of A (in some order). The minimal authorized sets of the access structure Γ^n_{Csi} are $\{p_i\} \cup A_i$.*

Theorem 7.8. *For every n, in any secret-sharing scheme realizing the n-party access structure Γ^n_{Csi},*

$$\sum_{p_j \in A} H(\{p_j\}) \geq \Omega(n) \cdot H(S).$$

In particular, the information ratio of every secret-sharing scheme realizing Γ^n_{Csi} is at least $\Omega(n/\log(n))$.

Proof. For the proof assume, without loss of generality, that $A = A_1, A_1, \ldots, A_{2^k} = \emptyset$ are ordered such that if $j < i$, then $A_j \not\subseteq A_i$ (this access structure is isomorphic to any access structure with a different order on the subsets). Clearly, $\{p_i\} \cup A_i \subseteq \{p_1, \ldots, p_i\} \cup A_i \in \Gamma^n_{\text{Csi}}$. Furthermore, $\{p_1, \ldots, p_{i-1}\} \cup A_i$ does not contain any minimal authorized set of Γ^n_{Csi} since $A_j \not\subseteq A_i$ for every $1 \leq j \leq i - 1$. Thus, $A_1, A_1, \ldots, A_{2^k}$ are an independent sequence of Γ^n_{Csi}, and by Theorem 7.6, $\sum_{p_j \in A} H(\{p_j\}) \geq (2^k - 1) \cdot H(S) = \Omega(n) \cdot H(S)$ and the information ratio of every scheme realizing Γ^n_{Csi} is at least $(2^k - 1)/|A| = \Omega(n/\log(n))$. □

Remark 7.9. The access structure Γ^n_{Csi} we defined in Definition 7.7 is from [43] and is similar to the access structure defined in [66]. Csirmaz [66] originaly proved the lower bound for the following access structure: For a given n, let k, A, and B be as defined in Definition 7.7. Let $A = A_1, A_1, \ldots, A_{2^k} = \emptyset$ be an ordering of the subsets of A such that if $j < i$, then $A_j \not\subseteq A_i$, as in the proof of Theorem 7.8 (e.g., if $j < i$, then $|A_j| > |A_i|$). The minimal authorized sets of Csirmaz's access structure are $\{p_1, \ldots, p_i\} \cup A_i$.

Beimel [16] showed how to use Theorem 7.6 to prove a lower bound of $\Omega(n^{1-1/(w-1)}/w)$ for w-hypergraph access structures, i.e., access structures in which the size of the minimal authorized sets are exactly w (where $3 \leq w \leq \log(n)$). Proving a similar lower bound for graphs (i.e., $w = 2$) is an open problem.

We next show how to strengthen Theorem 7.8 and show that there exists an access structure in which the shares of many parties have to be long. By Theorem 7.8, in Γ^n_{Csi} there is a small set A of size $O(\log(n))$ such that the sum of the entropies of the shares given to the parties in the set is $\Omega(n)H(S)$. In Definition 7.10, we construct a similar access structure that has many copies of A and one copy of B and in Theorem 7.11 we prove the lower bound on its total share size.

Definition 7.10 (The Access Structure Γ^n_{CsiTot})**.** *Fix $n \in \mathbb{N}$ and let k be the largest integer such that $2^k \leq n/2$. Let $B = \{p_1, \ldots, p_{2^k}\}$ and $A^t = \{p_{2^k+tk+1}, \ldots, p_{2^k+(t+1)k}\}$ for $0 \leq t \leq \lfloor n/2k \rfloor - 1$, and $A^t_1, A^t_2, \ldots, A^t_{2^k}$ be all the subsets of A^t (in some order). The minimal sets of Γ^n_{CsiTot} are $\{p_i\} \cup A^t_i$ for $1 \leq i \leq 2^k$ and $0 \leq t \leq \lfloor n/2k \rfloor - 1$.*

Theorem 7.11 ([67])**.** *For every n, the* total *information ratio of any secret-sharing scheme realizing Γ^n_{CsiTot} is $\Omega(n^2/\log(n))$.*

Proof. For every t, the access structure Γ^n_{CsiTot} restricted to the parties in $\{p_1, \ldots, p_{2^k}\} \cup A^t$ is isomorphic to the access structure $\Gamma^{k+2^k}_{\text{Csi}}$ (where $k + 2^k = \Omega(n)$). Thus, by Theorem 7.8,

$$\sum_{p_j \in A^t} H(\{p_j\}) \geq (2^k - 1)H(S) = \Omega(n)H(S).$$

As the sets A^t are disjoint,

$$\begin{aligned}\sum_{j=1}^{n} H(\{p_j\}) &\geq \sum_{\ell=0}^{\lfloor n/2k \rfloor - 1} \sum_{p_j \in A^t} H(\{p_j\}) \geq \left(\frac{n}{2k} - 1\right)\left(2^k - 1\right) H(S) \\ &= \Omega(n^2/\log(n))H(S).\end{aligned}$$

Thus, the total information ratio of every secret-sharing scheme realizing Γ^n_{CsiTot} is $\Omega(n^2/\log(n))$. □

7.4 The Framework for Proving Lower Bounds via Entropy and Its Limitations

Theorem 7.3 translates the question of proving lower bounds on the shares size of secret-sharing schemes realizing Γ to finding a minimum of a linear program, where for every set $A \subseteq \{p_1, \cdots, p_n\}$ we have a

variable and for every pair of sets we have one inequality. Note that the exact inequalities depend on the access structure (e.g., for two sets A, B such that $A \not\subseteq B$ and $B \not\subseteq A$ we use strong submodularity or submodularity). To bound the total share size we would like to minimize $\sum_{1 \leq i \leq n} H(\{p_i\})$.[20]

The lower bounds we described on the size of shares in secret-sharing schemes are implied by Theorem 7.3. In other words, they only use the so-called Shannon information inequalities (i.e., the fact that the conditional mutual information is non-negative). In 1998, new information inequalities were discovered by Zhang and Yeung [188]. Other information inequalities were discovered since, e.g. [126, 78, 134, 79, 103]. In particular, there are infinitely many independent information inequalities in 4 variables [134]. See [187] for a book on this subject. Beimel et al. [26] used non-Shannon inequalities to prove lower bounds on the share size. Each non-Shannon inequality adds many new inequalities to the linear program (for different subsets of the parties). Again, such inequalities also add strong inequalities, depending on which sets are in the access structure. This results in a huge linear program. For an access structure with few parties (e.g., 4,5, or 6), it can be solved using linear programming software (see, for example, [136, 84, 129, 147]).

Csirmaz [66] in 1994 proved that the linear program with Shannon-type inequalities (i.e., Theorem 7.3) cannot prove a lower bound of $\omega(n)$ on the information ratio. That is, Csirmaz's lower bound is nearly the best bound that can be proved using Shannon inequalities (up to a $\log(n)$ factor). Beimel and Orlov [27] proved that all information inequalities with 4 or 5 variables and some known information inequalities in more than 5 variables cannot prove a lower bound of $\omega(n)$ on the information ratio of secret-sharing schemes. Martín, Padró, and Yang [131] proved that all information inequalities on a bounded number of variables can only provide lower bounds that are polynomial on the number of parties. Thus, new information inequalities with many variables should be found if one wants to improve the lower bounds using this framework.

Remark 7.12. Farràs et al. [82] described a new method to derive inequalities in the linear programming technique, that is, they obtained non-Shannon-type bounds without using information inequalities explicitly. They derived better lower bounds on the information ratio of secret-sharing schemes of specific access structures using the linear program generated by their new method. More importantly, this method may bypass the limitations of [66, 26, 131]. However, they were not able to use their method to obtain better asymptotic lower bounds than [66, 67].

7.5 Lower Bounds for Linear Secret Sharing for Almost All Access Structures

For linear secret-sharing schemes, we can prove much stronger lower bounds than for general secret-sharing schemes, i.e., we can prove exponential lower bounds. Such bounds are known for almost all access structures [11, 22, 159, 19] and for explicit access structures [150].

[20] To bound the maximal share size we add another variable M, add the inequalities $H(\{p_i\}) \leq M$ for every $1 \leq i \leq n$, and minimize M.

Babai et al. [11] proved that for every prime-power q almost all n-party access structures require share size of at least $2^{0.5n-o(n)}$ in any linear secret-sharing scheme over $\mathbb{F}_q$. Theorem 7.14, proven below, is from [22] and is stronger than the result of [11]; it shows that almost all n-party access structures require share size of at least $2^{0.5n-o(n)}$ in any linear secret-sharing scheme over all finite fields (that is, the order of quantifiers is changed compared to [11]). The results follow by counting the number of monotone span programs, which by Claim 4.12 are equivalent to linear secret-sharing schemes. To count monotone span programs we need the following claim.

Claim 7.13. *Let* $\mathsf{MSP} = \left\langle \mathbb{F}_q, M, \rho \right\rangle$ *be a monotone span program of size* α *accepting a non-empty access structure. Then, without loss of generality, we can assume that* M *has at most* α *columns.*

Proof. W.l.o.g., assume that MSP accepts at least one input. If M contains more than α columns, then the columns of M are dependent. Restrict M to a matrix M' by taking a basis of the columns containing the first column. Notice that for every set of parties B the matrix M_B spans $\mathbf{e_1}$ if and only if M'_B spans $\mathbf{e_1}$. □

The next theorem proves exponential lower bounds on the share size of linear secret-sharing schemes for almost all access structures. The proof is somewhat technical since we need to prove the lower bound simultaneously for all fields.

Theorem 7.14. *For almost all* n*-party access structures* Γ *for all fields* $\mathbb{F}_q$ *the total share size of every linear secret-sharing scheme over* $\mathbb{F}_q$ *realizing* Γ *is at least*

$$\Omega\left(\frac{2^{n/2}}{n^{1/4}}\right).$$

Proof. If we share a secret using a linear secret-sharing scheme over $\mathbb{F}_q$ in which the shares contain α field elements, then the total size of the shares is $\alpha \cdot \log(q)$. For the total share size to be less than $2^{n/2}$, it must be that $q \leq 2^{2^{n/2}}$ (otherwise, each share contains at least $\log(q) \geq 2^{n/2}$ bits).

We next provide an upper bound on the number of linear secret-sharing schemes with share size at most $2^{n/2}$. For every finite field $\mathbb{F}_q$ and an integer $\alpha > \log(n)$, there are at most $n^\alpha \cdot q^{\alpha^2} < 2^{2\alpha^2 \log(q)}$ monotone span programs of size α, where the first term is an upper bound on the number of labeling functions ρ and the second term is an upper bound on the number of matrices M with α rows and α columns (by Claim 7.13 this is the number of columns).[21] Thus, the number of monotone span programs that can be used to construct a secret-sharing scheme with total share size at most $\beta = a \log(q)$ is at most

$$\sum_{q \,:\, q \leq 2^{2^{n/2}} \text{ is a prime-power}} 2^{2(\beta/\log(q))^2 \log(q)} \leq 2^{2^{n/2}} 2^{2\beta^2}.$$

The number of n-party access structures is at least $2^{\binom{n}{n/2}} \gg 2^{2^n/(2\sqrt{n})}$ (for each set of parties of size $n/2$ we choose if it is a minimal authorized set or unauthorized).

[21] By adding zero columns and rows we can assume that the number of columns and rows is exactly α.

If $2^{2^n/(2\sqrt{n})}$ access structures with n parties have a linear secret-sharing scheme with total share size at most β, then it must hold that

$$2^{2^{n/2}+2\beta^2} \geq 2^{2^n/(2\sqrt{n})},$$

which, in particular, implies that

$$\beta > \sqrt{\frac{1}{2}\left(\frac{2^n}{2\sqrt{n}} - 2^{n/2}\right)} > \frac{2^{n/2}}{3n^{1/4}}.$$

To conclude, for all n-party access structures, but at most $2^{2^{n/2}/(2\sqrt{n})}$ access structures, the total share size in every linear secret-sharing scheme realizing them is at least $\frac{2^{n/2}}{3n^{1/4}}$. □

Theorem 7.14 gives a lower bound on the share size of linear secret-sharing schemes; however, it does not give a lower bound on the information ratio of linear secret-sharing schemes for long secrets. Beimel and Farràs [19] (using a result of Nelson [141]) proved such a lower bound (improving on [11, 159]). This lower bound proves that almost all access structures require long shares in linear secret-sharing schemes over all fields simultaneously.

Theorem 7.15. *For almost all n-party access structures Γ for all fields $\mathbb{F}_q$ the total information ratio of every linear secret-sharing scheme over $\mathbb{F}_q$ realizing Γ is at least*

$$\Omega\left(2^{n/3-o(n)}\right).$$

7.6 Lower Bounds for Linear Secret Sharing for Explicit Access Structures

We next discuss lower bounds for linear secret-sharing schemes for explicit access structures, that is, for access structures that have a small representation in a natural representation model. Such lower bounds were proved in a sequence of works [23, 11, 89, 90, 158, 149, 150], resulting in an exponential lower bound [150].

Theorem 7.16 ([150])**.** *For every n, there exists an explicit access structure such that the information ratio of every linear secret-sharing scheme realizing it is $2^{\Omega(n)}$.*

The proof of the exponential lower bounds are beyond this monograph; they can be found in [150, 157]. We present a proof from [90] of a lower bound of $n^{\Omega(\log(n))}$ for an explicit access structure. Recall that linear secret-sharing schemes are equivalent to monotone span programs (see Claim 4.12); we prove the lower bounds for secret-sharing schemes by proving lower bounds for monotone span programs. We start with a simple observation.

Observation 7.17. *Let Γ be a (monotone) access structure. Let $B \in \Gamma$ and $C \subseteq \{p_1, \ldots, p_n\}$ such that $\{p_1, \ldots, p_n\} \setminus C \notin \Gamma$. Then, $B \cap C \neq \emptyset$.*

The observation follows from the fact that if $B \cap C = \emptyset$, then $B \subseteq \{p_1, \ldots, p_n\} \setminus C$, contradicting the fact that $B \in \Gamma$ and $\{p_1, \ldots, p_n\} \setminus C \notin \Gamma$.

To prove the lower bound, Gál and Pudlák [90] chose a subset of the unauthorized sets that satisfies some properties; they use this subset to construct a matrix over $\mathbb{F}$, and prove that the rank of the matrix over $\mathbb{F}$ is a lower bound on the size of every monotone span program realizing Γ.

Let $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ be the collection of minimal authorized sets in Γ and $\mathcal{C} = \{\langle C_{1,0}, C_{1,1}\rangle, \langle C_{2,0}, C_{2,1}\rangle, \ldots \langle C_{t,0}, C_{t,1}\rangle\}$ be a collection of pairs of sets of parties such that $\{p_1, \ldots, p_n\} \setminus (C_{j,0} \cup C_{j,1}) \notin \Gamma$ for every $1 \leq j \leq t$. By Observation 7.17, $B_i \cap (C_{j,0} \cup C_{j,1}) \neq \emptyset$ for every i, j, that is, at least one of the following conditions hold: (1) $B_i \cap C_{j,0} \neq \emptyset$, (2) $B_i \cap C_{j,1} \neq \emptyset$. To prove the lower bound, Gál and Pudlák use a collection $\mathcal{C}$ such that, for every i, j, *exactly* one of the above conditions hold.

Definition 7.18. *Let $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ be the collection of minimal authorized sets in Γ. We say that a collection $\mathcal{C} = \{\langle C_{1,0}, C_{1,1}\rangle, \langle C_{2,0}, C_{2,1}\rangle, \ldots \langle C_{t,0}, C_{t,1}\rangle\}$ of pairs of sets satisfies the* unique intersection *property for Γ if*

1. *For every $1 \leq j \leq t$, $\{p_1, \ldots, p_n\} \setminus (C_{j,0} \cup C_{j,1}) \notin \Gamma$.*
2. *For every $1 \leq i \leq \ell$ and every $1 \leq j \leq t$, exactly one of the following conditions hold (1) $B_i \cap C_{j,0} \neq \emptyset$, (2) $B_i \cap C_{j,1} \neq \emptyset$.*

Example 7.19. Consider the access structure with ten parties $\{p_1, \ldots, p_{10}\}$ and six minimal authorized sets $\{p_1, p_2, p_5\}$, $\{p_1, p_3, p_6\}$, $\{p_1, p_4, p_7\}$, $\{p_2, p_3, p_8\}$, $\{p_2, p_4, p_9\}$, and $\{p_3, p_4, p_{10}\}$. We next define a collection $\mathcal{C}$ satisfying the unique intersection property for Γ, where $\mathcal{C}$ is $\langle \{p_1, p_2\}, \{p_{10}\}\rangle$, $\langle \{p_1, p_3\}, \{p_9\}\rangle$, $\langle \{p_1, p_4\}, \{p_8\}\rangle$, $\langle \{p_2, p_3\}, \{p_7\}\rangle$, $\langle \{p_2, p_4\}, \{p_6\}\rangle$, and $\langle \{p_3, p_4\}, \{p_5\}\rangle$.

It can be seen that $\mathcal{C}$ satisfies Item 1 of Definition 7.18. For example, the set $T = \{p_1, \ldots, p_{10}\} \setminus (\{p_1, p_2\} \cup \{p_{10}\}) = \{p_3, p_4, \ldots, p_9\}$ is unauthorized. Furthermore, $\mathcal{C}$ satisfies Item 2 of Definition 7.18. Consider, e.g., $\{p_1, p_3, p_6\} \in \mathcal{B}$ and $\langle \{p_1, p_2\}, \{p_{10}\}\rangle \in \mathcal{C}$. In this case $\{p_1, p_3, p_6\} \cap \{p_1, p_2\} \neq \emptyset$ while $\{p_1, p_3, p_6\} \cap \{p_{10}\} = \emptyset$.

Theorem 7.20 ([90])**.** *Let $\mathcal{C}$ be a collection satisfying the unique intersection property for Γ and define an $\ell \times t$ matrix D, where $D_{i,j} = 0$ if $B_i \cap C_{j,0} \neq \emptyset$ and $D_{i,j} = 1$ if $B_i \cap C_{j,1} \neq \emptyset$. Then, the size of every monotone span program over $\mathbb{F}$ accepting Γ is at least* $\mathrm{rank}_{\mathbb{F}}(D)$.

Example 7.21. The matrix D defined for the set $\mathcal{C}$ of Example 7.19 is the full-rank matrix described below. For example, the minimal authorized set $\{p_1, p_2, p_5\}$ intersects the first set in $\langle \{p_1, p_2\}, \{p_{10}\}\rangle$, so $D_{1,1} = 0$. Similarly, $\{p_1, p_2, p_5\}$ intersects the second set in $\langle \{p_3, p_4\}, \{p_5\}\rangle$, so $D_{1,6} = 1$.

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

Proof of Theorem 7.20. Let $\mathsf{MSP} = \langle \mathbb{F}, M, \rho \rangle$ be a monotone span program accepting Γ (as defined in Definition 4.7), and denote the size of MSP (i.e., the number of rows in M) by m. We will construct two matrices L and R, where L has m columns and R has m rows such that $D = LR$. Thus, $\mathrm{rank}_{\mathbb{F}}(D) \leq \mathrm{rank}_{\mathbb{F}}(L) \leq m$, i.e., the size of MSP is at least $\mathrm{rank}_{\mathbb{F}}(D)$.

Fix any i such that $1 \leq i \leq \ell$. Since $B_i \in \Gamma$, the rows in M labeled by the parties in B_i span the vector $\mathbf{e_1}$, that is, there exists a vector $\mathbf{v_i}$ such that $\mathbf{v_i}M = \mathbf{e_1}$ and the non-zero coordinates of $\mathbf{v_i}$ are only in rows labeled by B_i (where the dth coordinate of $\mathbf{v_i}$ is labeled by $\rho(d)$).

Fix any j such that $1 \leq j \leq t$, and let $T_j = \{p_1, \ldots, p_n\} \setminus (C_{j,0} \cup C_{j,1})$. Since $T_j \notin \Gamma$, the rows in M labeled by the parties in T_j do not span the vector $\mathbf{e_1}$. As explained in Section 4.4, there exists a vector $\mathbf{w_j}$ such that $M_{T_j}\mathbf{w_j} = \mathbf{0}$ and $\mathbf{e_1} \cdot \mathbf{w_j} = 1$. Let $\mathbf{y_j} \stackrel{\text{def}}{=} M\mathbf{w_j}$. Note that all coordinates in $\mathbf{y_j}$ labeled by the parties in T_j are zero. Furthermore, for every i, j,

$$\mathbf{v_i}\mathbf{y_j} = \mathbf{v_i}(M\mathbf{w_j}) = (\mathbf{v_i}M)\mathbf{w_j} = \mathbf{e_1} \cdot \mathbf{w_j} = 1.$$

We next modify the vectors $\mathbf{y_1}, \ldots, \mathbf{y_t}$ to vectors $\mathbf{z_1}, \ldots, \mathbf{z_t}$ such that $\mathbf{v_i}\mathbf{z_j} = D_{i,j}$ for every i, j. Let $\mathbf{z_j}$ be the column vector achieved from $\mathbf{y_j}$ by replacing all coordinates in $\mathbf{y_j}$ labeled by parties in $C_{j,0}$ with 0. Thus, the non-zero coordinates in $\mathbf{z_j}$ are labeled by parties in $C_{j,1}$ and the non-zero coordinates in both $\mathbf{v_i}$ and $\mathbf{z_j}$ are labeled by parties in $B_i \cap C_{j,1}$ (since all non-zero coordinates in $\mathbf{v_i}$ are labeled by B_i). Hence,

- If $B_i \cap C_{j,0} \neq \emptyset$, then $B_i \cap C_{j,1} = \emptyset$. In this case, $D_{i,j} = 0$ and $\mathbf{v_i}$ and $\mathbf{z_j}$ do not share non-zero coordinates, thus, $\mathbf{v_i} \cdot \mathbf{z_j} = 0 = D_{i,j}$.
- If $B_i \cap C_{j,1} \neq \emptyset$, then $D_{i,j} = 1$ and all coordinates in $\mathbf{v_i}$ labeled by $C_{j,0}$ are zero (since $B_i \cap C_{j,0} = \emptyset$), thus, $\mathbf{v_i} \cdot \mathbf{z_j} = \mathbf{v_i} \cdot \mathbf{y_j} = 1 = D_{i,j}$.

Define a matrix L, where the ith row in L is $\mathbf{v_i}$, and a matrix R, where the jth column of R is $\mathbf{z_j}$. We claim that $D = LR$ since $D_{i,j} = \mathbf{v_i} \cdot \mathbf{z_j}$. As L has m columns, $\mathrm{rank}_{\mathbb{F}}(D) = \mathrm{rank}_{\mathbb{F}}(LR) \leq \mathrm{rank}_{\mathbb{F}}(L) \leq m$. In other words, the rank of D is at most the size of the smallest monotone span program accepting Γ. □

We next present a construction of an access structure for which we can prove an $n^{\Omega(\log(n))}$ lower bound using Theorem 7.20. An undirected bipartite graph $G = (U, V, E)$ has the isolated neighbor property for t if for every two disjoint sets $A_1, A_2 \subset U$ such that $|A_1| = |A_2| = t$, there exists a vertex $v \in V$ such that $(u_1, v) \in E$ for every $u_1 \in A_1$ and $(u_2, v) \notin E$ for every $u_2 \in A_2$, that is, v is a neighbor of every vertex in A_1 and is isolated from every vertex in A_2.

For a set $A \subset U$ define $N(A) \stackrel{\text{def}}{=} \{v : \forall_{u \in A}(u, v) \in E\}$, that is, a vertex is in $N(A)$ if it is a neighbor of all vertices in A. Let $G = (U, V, E)$ be an undirected bipartite graph satisfying the isolated neighbor property for t, where the vertices of the graph are parties, i.e., $U \cup V = \{p_1, \ldots, p_n\}$. We define an access structure $\mathcal{N}_G$ with $|U| + |V|$ parties whose minimal authorized sets are the sets $A \cup N(A)$ where $A \subset U$ and $|A| = t$.

Example 7.22. Consider the graph described in Figure 7.1. This is a trivial graph satisfying the isolated neighbor property for $t = 2$, i.e., for every pair of vertices, there exists a unique vertex that is adjacent only to

them. For example, consider the disjoint sets $\{p_1, p_2\}$ and $\{p_3, p_4\}$; vertex p_5 is a neighbor of all the vertices in the first set, while it is not a neighbor of any vertex in the second set. The access structure $\mathcal{N}_G$ defined for this graph is the access structure defined in Example 7.19.

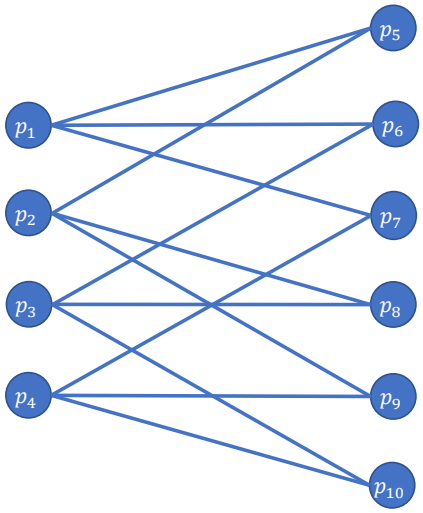


Figure 7.1: An example of a graph satisfying the isolated neighbor property for $t = 2$.

Lemma 7.23. *If $G = (U, V, E)$ has the isolated neighbor property for t, then the size of every monotone span program over $\mathbb{F}$ accepting $\mathcal{N}_G$ is at least $\binom{|U|-1}{t}$.*

Proof. We prove the lemma using Theorem 7.20. We take $\mathcal{C}$ to be all the pairs C_0, C_1, where $C_0 \subset U$ such that $|C_0| = t$ and $C_1 = \left\{ v \in V : \forall_{u \in C_0} (u, v) \notin E \right\}$, that is, C_1 contains all vertices that are not neighbors of any vertex in C_1. We first claim that the collection $\mathcal{C}$ satisfies the unique intersection property for Γ:

- Let $(C_0, C_1) \in \mathcal{C}$ and $T = \{p_1, \ldots, p_n\} \setminus (C_0 \cup C_1)$. We need to show that $T \notin \Gamma$, that is, T does not contain any minimal authorized set. Consider any minimal authorized set $A \cup N(A)$, where $A \subseteq U \cap T$ and $|A| = t$; clearly, $A \neq C_0$ (as $A \subseteq T$ and $C_0 \cap T = \emptyset$). As $|A| = |C_0| = t$, by the isolated neighbor property there is a vertex $v \in V$ such that $v \in N(A)$ and v is not a neighbor of any vertex in C_0, thus, $v \in C_1$ by the definition of C_1, that is, $v \notin T$. In other words, T does not contain any minimal authorized set $A \cup N(A)$.

- Let $A \cup N(A) \in \mathcal{N}_G$ and $\langle C_0, C_1 \rangle \in \mathcal{C}$. First notice that $(A \cup N(A)) \cap C_0 = A \cap C_0$ and $(A \cup N(A)) \cap C_1 = N(A) \cap C_1$. Assume that $A \cap C_0 \neq \emptyset$, and let $u \in A \cap C_0$. Thus, every $v \in N(A)$ is a neighbor of u. However, every vertex in C_1 is not a neighbor of u, and $(A \cup N(A)) \cap C_1 = N(A) \cap C_1 = \emptyset$.

Thus, by Theorem 7.20, the size of every monotone span program accepting Γ is at least $\text{rank}_{\mathbb{F}}(D)$. In this case, for every A, C_0 such that $|A| = |C_0| = t$, the entry corresponding to $A \cup N(A)$ and $\langle C_0, C_1 \rangle$ is zero if

$A \cap C_0 \neq \emptyset$ and is one otherwise. That is, D is the (n,t)-disjointness matrix, which has full rank over every field $\mathbb{F}$ such that $|U|-2t+1 \neq 0$ in the field $\mathbb{F}$ (see, e.g., [120, Example 2.12]).If $|U|-2t+1=0$ in the field $\mathbb{F}$, consider the set U' containing the first $|U|-1$ vertices of the U. The matrix restricted to the rows and columns labeled by subsets of size t contained in U' has full rank (since $|U'|-2t+1=|U|-1-2t+1=-1$ over $\mathbb{F}$). Thus, the rank of D over all fields is at least $\binom{|U|-1}{t}$. □

As there exist n-vertex graphs that satisfy the isolated neighbor property for $t=\Omega(\log(n))$, e.g., the Paley Graph [3], we derive the promised lower bound.

Theorem 7.24. *For every n, there exists an n-party access structure $\Gamma_{\mathrm{Paley},n}$ such that every monotone span program over any field accepting it has size $n^{\Omega(\log(n))}$.*

As monotone span programs are equivalent to linear secret-sharing schemes [111, 14], the same lower bound applies to linear secret-sharing schemes.

Corollary 7.25. *For every n, there exists an n-party access structure $\Gamma_{\mathrm{Paley},n}$ such that the information ratio of every linear secret-sharing scheme realizing it is $n^{\Omega(\log(n))}$.*

In multilinear secret-sharing schemes as defined in Definition 4.14, the secret can be a vector of elements over $\mathbb{F}$, which can reduce the information ratio. Beimel et al. [17] proved that the above lower bound also holds for multilinear secret-sharing schemes, obtaining the best lower bound for multilinear secret-sharing schemes (unlike linear secret-sharing schemes, where an exponential lower bound is known).

Corollary 7.26. *For every n, there exists an n-party access structure $\Gamma_{\mathrm{Paley},n}$ such that the information ratio of every* multilinear *secret-sharing scheme realizing it is $n^{\Omega(\log(n))}$.*

Chapter 8

Ideal Secret Sharing

This chapter studies the most efficient secret-sharing schemes and the access structures that they can realize. By a lower bound of Karnin et al. [112] (see Lemma 7.1), in every secret-sharing scheme the size of the share of each non-redundant party is at least the size of the secret. A secret-sharing is ideal if the size of each share is this minimal possible size, i.e., the size of the share of each party is exactly the size of the secret. For example, Shamir's t-out-of-n secret-sharing scheme [163] (see Section 2.2) is ideal when the size of the domain of secrets is a prime-power $q > n$. An access structure is ideal if it has an ideal scheme over some finite domain of secrets. For example, the t-out-of-n access structure is ideal, while the access structure $\Gamma_{\sqcap}$ described in Example 4.16 is not ideal [36] (see Theorem 7.4). Benaloh and Leichter [36], Simmons [165], and Brickell [49] constructed ideal schemes for some access structures, i.e., for hierarchical access structures. Brickell and Davenport [50] showed an interesting connection between ideal access structures and matroids. Matroids are an abstraction and generalization of linear independence in vector spaces and spanning trees in graphs; they were defined by Whitney in 1935 [184].

Brickell and Davenport provided a necessary condition for being ideal and a sufficient condition using ports of matroids, which are defined in Section 8.1 (this chapter is self contained and no background on matroid is needed). These conditions provide a partial characterization of ideal access structures as there is a gap between these two conditions. The conditions are informally stated below.

- If an access structure is ideal then it is a matroid port.
- If an access structure is a matroid port of a representable matroid, then the access structure is ideal.

We remark that ideal secret-sharing schemes have been defined and studied in other areas of research under different names, i.e., Matúš [133] studied them using the name probabilistic representation and Simonis and Ashikhmin [167] studied them using the term almost affine codes.

8.1 Definition of Ideal Secret Sharing and Background on Matroids

We start with the definition of ideal access structures.

Definition 8.1 (Ideal Access Structures)**.** *A secret-sharing scheme with domain of secrets S is* ideal *if the domain of shares of each party is S. An access structure Γ is* ideal *if there exists an ideal secret-sharing scheme realizing it over some finite domain of secrets.*

A party is *redundant* in an access structure if there is no minimal authorized set that contains it. If a party is redundant, then it does not need to get a share. Thus, in the following discussion we only consider access structures without redundant parties.

A matroid is an axiomatic abstraction of linear independence. There are several equivalent axiomatic systems to describe matroids: by independent sets, by bases, by circuits, or, as done here, by the rank function. For more background on matroid theory the reader is referred to [183, 144].

Definition 8.2 (Matroids [184])**.** *A matroid $\mathcal{M} = \langle V, \mathrm{rank} \rangle$ is a finite set V and a function* rank $: 2^V \to \mathbb{N}$ *satisfying the following three axioms:*

Non-negativity and Boundness. $0 \leq \mathrm{rank}(X) \leq |X|$ *for every* $X \subseteq V$.

Monotinicity. $\mathrm{rank}(X) \leq \mathrm{rank}(Y)$ *for every* $X \subseteq Y \subseteq V$.

Sub-modularity. $\mathrm{rank}(X \cup Y) + \mathrm{rank}(X \cap Y) \leq \mathrm{rank}(X) + \mathrm{rank}(Y)$.

The elements of V are called points*, or simply* elements*, and the function* rank *is called the rank function.*

Example 8.3. Let V be a set of vectors in $\mathbb{F}^k$ for some field $\mathbb{F}$ and $\mathrm{rank}_{\mathrm{LIN}}(X)$ be the linear-algebraic rank of X (i.e., the dimension of the linear space spanned by the vectors in X). Then, $\langle V, \mathrm{rank}_{\mathrm{LIN}} \rangle$ is a matroid since

$$\begin{aligned} \mathrm{rank}_{\mathrm{LIN}}(X \cup Y) + \mathrm{rank}_{\mathrm{LIN}}(X \cap Y) &\leq \mathrm{rank}_{\mathrm{LIN}}(\mathrm{span}(X) \cup \mathrm{span}(Y)) + \mathrm{rank}_{\mathrm{LIN}}(\mathrm{span}(X) \cap \mathrm{span}(Y)) \\ &= \mathrm{rank}_{\mathrm{LIN}}(X) + \mathrm{rank}_{\mathrm{LIN}}(Y). \end{aligned} \tag{8.1}$$

A matroid defined by such a set of vectors is called representable (see Definition 8.10).

Notice that (8.1) can be a strict inequality. For example, let $X = \{e_1\}$ and $Y = \{e_2\}$; in this case, $\mathrm{rank}_{\mathrm{LIN}}(X \cap Y) = \mathrm{rank}_{\mathrm{LIN}}(\emptyset) = 0$.

Definition 8.4 (Independent Sets, Dependent Sets, and Circuits)**.** *A subset of $X \subseteq V$ is* independent *in a matroid $\mathcal{M}$ if* $\mathrm{rank}(X) = |X|$. *A subset of $X \subseteq V$ is* dependent *in a matroid $\mathcal{M}$ if* $\mathrm{rank}(X) < |X|$. *Since* $\mathrm{rank}(X)$ *is an integer, a set X is dependent if and only if* $\mathrm{rank}(X) \leq |X| - 1$. *A subset of $C \subseteq V$ is a* circuit *if it is a minimal dependent set; i.e.,* $\mathrm{rank}(C) = |C| - 1$ *and* $\mathrm{rank}(C \setminus \{a\}) = |C \setminus \{a\}| = |C| - 1$ *for every* $a \in C$.

Example 8.5. Let $G = (V, E)$ be an undirected simple graph and for $X \subseteq E$ let $\mathrm{rank}_G(X)$ be the number of edges in a spanning forest of $G_X = (V, X)$.[22] It can be proved that (E, rank_G) is a matroid, where $\mathrm{rank}_G(X)$

[22] A forest is a graph without cycles. A spanning forest $H = (V, F)$ of $G = (V, E)$ is a maximal sub-graph of G without cycles, that is $H = (V, F)$ is a forest such that $F \subseteq E$ and $(V, F \cup \{e\})$ contains a cycle for every $e \in E \setminus F$. The number of edges in every spanning forest of $G = (V, E)$ is the same (i.e., $|V| - \ell$, where ℓ is the number of connected components of G).

is $|V|$ minus the number of connected components in the graph (V, X). Its independent sets are forests, its dependent sets are sets that contain cycles, and its circuits are the simple circuits of the graphs, i.e., simple cycles. The matroid $\langle E, \mathrm{rank}_G \rangle$ is called a graphic matroid.

Definition 8.6 (Connected Matroid)**.** *A matroid is* connected *if for every pair of distinct elements x and y there is a circuit containing x and y.*

We next define ports of a matroid, which is the key notion for studying ideal access structures.

Definition 8.7 (Ports of a Matroid)**.** *Let $\mathcal{M} = \langle V, \mathrm{rank} \rangle$ be a connected matroid, $\mathcal{C}$ be the circuits of the matroid, and $p_0 \in V$. The* port of the matroid $\mathcal{M}$ at point p_0 *is the access structure Γ on $P = V \setminus \{p_0\}$ whose minimal authorized sets are*

$$\left\{ A \subseteq V \setminus \{p_0\} \,:\, A \cup \{p_0\} \in \mathcal{C} \right\}.$$

That is, a set A is a minimal authorized set of Γ if by adding p_0 to it, it becomes a circuit of $\mathcal{M}$.

Example 8.8. Let $\mathcal{M}$ be the graphic matroid of the complete graph G. Let Γ be the port of $\mathcal{M}$ at the point (i.e., edge) (v_1, v_m). A minimal set in Γ is a set A such that $A \cup \{(v_1, v_m)\}$ is a simple cycle in G, i.e., A is a simple path between v_1 and v_m. In other words, $\Gamma = \Gamma_{\mathrm{ustcon}}$, where Γ_{ustcon} is the connectivity access structure defined in Section 4.1.

The following theorem, whose proof can be found in [144, Theorem 4.3.2], states that an access structure can be a port of at most one matroid.

Theorem 8.9. *Let Γ be an access structure without redundant parties. If Γ is a port of a matroid $\mathcal{M}$ at point p_0, then $\mathcal{M}$ is uniquely determined by Γ, that is, Γ can be a port of at most one matroid at p_0. Furthermore, if Γ is a port of a matroid $\mathcal{M}$, then $\mathcal{M}$ is a connected matroid.*

8.2 Ideal Secret Sharing from Representable Matroids

We show a sufficient condition of Brickell and Davenport [50] for an ideal access structure; namely, if an access structure is a port of a representable matroid (over some finite field), then it is ideal.[23] The construction of a secret-sharing scheme from a representation of a matroid is a special case of the monotone span program construction of [111] (see Chapter 6). We next formally define representable matroids.

Definition 8.10. *A matroid $\mathcal{M} = (V, \mathrm{rank})$ is* representable *over a field $\mathbb{F}$ if there exists a rank-preserving mapping from the points of the matroid into the set of vectors of a vector space over the field. In other words, there exist k and a mapping $\phi : V \to \mathbb{F}^k$ such that for every $A \subseteq V$:*

$$\mathrm{rank}(A) = \mathrm{rank}_{\mathrm{LIN}}(\phi(A)).$$

[23] In [50], it is claimed that the matroid can be representable over a near field; however, Simonis and Ashikhmin [167] have shown an example in which the construction from a near field is incorrect.

The above requirement of ϕ is equivalent to requiring that a set $A \subseteq V$ is a dependent set of the matroid if and only if $\phi(A)$ is linearly dependent.

Theorem 8.11 ([50])**.** *If an access structure Γ without redundant parties is a port of a matroid $\mathcal{M}$ representable over a finite field $\mathbb{F}_q$, then Γ can be realized by an ideal secret-sharing scheme with a domain of secrets of size q.*

Proof. Let $\phi : V \to \mathbb{F}^k$ be a representation of $\mathcal{M} = \left\langle V = P \cup \left\{p_0\right\}, \text{rank}\right\rangle$. By changing the basis of $\left\{\phi(p_i) : p_i \in P \cup \left\{p_0\right\}\right\}$, we can assume that $\phi(p_0) = \mathbf{e_1}$ and consider the monotone span program MSP with n rows, where for every $1 \leq i \leq n$ there is a row $\phi(p_i)$ labeled by p_i. Since Γ is a port of $\mathcal{M}$ at p_0, a set A is a minimal authorized set of Γ if and only if $A \cup \left\{p_0\right\}$ is a minimal dependent set in $\mathcal{M}$ if and only if $\left\{\phi(p_i) : p_i \in A \cup \left\{p_0\right\}\right\}$ is a minimal linearly-dependent set if and only if $\left\{\phi(p_i) : p_i \in A\right\}$ is a minimal set that spans $\mathbf{e_1}$ if and only if A is a minimal set accepted by the monotone span program MSP. We conclude that MSP is a monotone span program accepting Γ, where every party labels one row. Thus, by Claim 4.9, there is an ideal secret-sharing scheme realizing Γ. □

A multilinear representation of a matroid $\mathcal{M} = \langle V, \text{rank}\rangle$ is a mapping $\phi : V \to (\mathbb{F}^k)^\ell$ for some $\ell \geq 1$ (i.e., each point is mapped to ℓ vectors) such that for every set $A \subseteq V$

$$\text{rank}(A) = \frac{\text{rank}_{\text{LIN}}(\phi(A))}{\ell}.$$

Note that a mapping $\phi : V \to (\mathbb{F}^k)^\ell$ is a multilinear representation of some matroid if and only if the above rank function is integral, i.e., $\text{rank}_{\text{LIN}}(\phi(A))$ is a multiple of ℓ for every set A.

Theorem 8.11 can be generalized to ports of multilinear representable matroids, using multi-target monotone span programs and multilinear secret-sharing schemes (see Section 4.6 for the definition of these notions).

Theorem 8.12. *If an access structure Γ without redundant parties is a port of a matroid $\mathcal{M}$ that has a multilinear representation over a finite field $\mathbb{F}_q$, then Γ can be realized by an ideal secret-sharing scheme with a domain of secrets of size q^ℓ for some integer $\ell \geq 1$.*

8.3 Matroids from Ideal Secret Sharing

The following fundamental result, proved by Brickell and Davenport [50], gives a necessary condition for an access structure to have an ideal secret-sharing scheme – the access structure is a port of a matroid. The proof we provide is from [119]; it defines the rank function of the matroid via the joint entropy of the collections of shares of an ideal secret-sharing scheme.

Theorem 8.13 ([50])**.** *If an access structure without redundant parties Γ is ideal, then Γ is a port of a matroid.*

Proof. Let $P = \{p_1, \ldots, p_0\}$ be the set of parties and p_0 be the dealer and let Π be an ideal secret-sharing scheme realizing Γ with a domain of secrets S. We consider its security according to Definition 3.7, where the secret is uniformly distributed in S. Let $\mathcal{S}_0 = \mathcal{S}$ be the random variable denoting the secret with uniform distribution and $\mathcal{S}_i$ be the random variable denoting the share of party p_i for $1 \leq i \leq n$. Recall that for a set $T \subseteq P \cup \{p_0\}$ we define $\mathcal{S}_T = \langle \mathcal{S}_i \rangle_{p_i \in T}$. We use the notation from Section 7.2, denoting $H(\mathcal{S}_A)$ by $H(A)$ and $H(\mathcal{S}_A, \mathcal{S})$ by $H(A, \mathcal{S})$. We start with two simple claims on these entropies for ideal secret-sharing schemes.

Claim 8.14. *If $C \notin \Gamma$ and $C \cup \{p_i\} \in \Gamma$, then $H(\{p_i\}|C) = H(\mathcal{S})$ and $H(\{p_i\}|\mathcal{S}, C) = 0$.*

Proof. By the definition of the conditional entropy (A.3) and by the security (3.6) and the correctness (3.5) of Π,

$$H(\mathcal{S}|C) + H(\{p_i\}|C) - H(\mathcal{S}, \{p_i\}|C) = H(\mathcal{S}|C) - H(\mathcal{S}|C, \{p_i\}) = H(\mathcal{S}) - 0 = H(\mathcal{S}). \tag{8.2}$$

By (8.2), the definition of the conditional entropy (A.3), the properties of conditional entropy (A.4), and the upper bound on the entropy (A.2),

$$\begin{aligned} H(\mathcal{S}) &= H(\mathcal{S}|C) + H(\{p_i\}|C) - H(\mathcal{S}, \{p_i\}|C) \\ &= H(\mathcal{S}|C) + H(\{p_i\}|C) - (H(\mathcal{S}|C) + H(\{p_i\}|\mathcal{S}, C)) \\ &= H(\{p_i\}|C) - H(\{p_i\}|\mathcal{S}, C) \\ &\leq H(\{p_i\}|C) \\ &\leq H(\{p_i\}) \leq H(\mathcal{S}). \end{aligned}$$

Thus, $H(\mathcal{S}) = H(\{p_i\}|C) - H(\{p_i\}|\mathcal{S}, C) = H(\{p_i\}|C)$, and the claim follows. □

Claim 8.15. *If $B \cup C \in \Gamma$ and $B \cup C \setminus \{p_i\} \notin \Gamma$ for every $p_i \in B$, then $H(B|C) = |B| \cdot H(\mathcal{S})$.*

Proof. Let $B = \{p_1, \ldots, p_t\}$. By Claim 8.14, $H(\{p_i\}|B \cup C \setminus \{p_i\}) = H(\mathcal{S})$. Thus, by the definition of conditional entropy (A.3) and by the properties of the conditional entropy (A.4),

$$H(B|C) = \sum_{i=1}^{t} H(\{p_i\}|C, \{p_1, \ldots, p_{i-1}\}) \geq \sum_{i=1}^{t} H(\{p_i\}|B \cup C \setminus \{p_i\}) = |B| \cdot H(\mathcal{S}). \tag{8.3}$$

On the other hand, by the definition of conditional entropy (A.3), (A.4), (A.2), and the fact that $\mathcal{S}$ is uniformly distributed in S,

$$H(B|C) = \sum_{i=1}^{t} H(\{p_i\}|C, \{p_1, \ldots, p_{i-1}\}) \leq \sum_{i=1}^{t} H(\{p_i\}) = |B| \cdot \log(|S|) = |B| \cdot H(\mathcal{S}). \tag{8.4}$$

□

We use the entropy of the shares of the set A to define a function rank_Π and prove that it is a rank function of a matroid. For every non-empty set $A \subseteq P \cup \{p_0\}$, define

$$\mathrm{rank}_\Pi(A) = H(A)/H(S),$$

and define $\mathrm{rank}_\Pi(\emptyset) = 0$. We will first prove the integrality of rank_Π; this is the hardest part of the proof. Then we will prove that $\langle P \cup \{0_0\}, \mathrm{rank}_\Pi \rangle$ is a matroid. Finally, we will prove that Γ is the port of the matroid at p_0.

Claim 8.16. $\mathrm{rank}_\Pi(A)$ *is an integer for every set* $A \subseteq P \cup \{p_0\}$.

Proof. First, note that $H(A \cup \{p_0\}) = H(S, A) = H(A) + H(S|A)$; by the correctness and security of Π, the entropy $H(S|A)$ is either zero or 1. Thus, we need to prove the integrality of $\mathrm{rank}_\Pi(A)$ only for subsets of P. Assume that rank_Π is not integral and let A be a minimal set such that $\mathrm{rank}_\Pi(A)$ is not an integer. Let $B \subseteq P \setminus A$ be a minimal set such that $B \notin \Gamma$ and $A \cup B \in \Gamma$; we next prove that such a set exists. If $A \in \Gamma$ we take $B = \emptyset$. Otherwise, such a set exists because Γ is connected: Take $p_i \in A$ and let C be a minimal authorized set containing p_i, thus $C \setminus \{p_i\}$ is unauthorized and $A \cup C = A \cup (C \setminus \{p_i\})$ is authorized; take $B \subseteq C \setminus A$ as a minimal set such that $A \cup B \in \Gamma$.

We consider two cases; in each case we derive a contradiction to the existence of a minimal set such that $\mathrm{rank}_\Pi(A)$ is not an integer.

Case I: There exists a $p_i \in A$ such that $A \cup B \setminus \{p_i\} \notin \Gamma$. By Claim 8.14, $H(\{p_i\} | A \cup B \setminus \{p_i\}) = H(S)$, thus by (A.4) and (A.2)

$$H(S) = H(\{p_i\} | A \cup B \setminus \{p_i\}) \leq H(\{p_i\} | A \setminus \{p_i\}) \leq H(\{p_i\}) \leq H(S),$$

i.e.,

$$H(\{p_i\} | A \setminus \{p_i\}) = H(S); \tag{8.5}$$

by (A.3) and (8.5)

$$H(A) = H(A \setminus \{p_i\}, \{p_i\}) = H(A \setminus \{p_i\}) + H(\{p_i\} | A \setminus \{p_i\}) = H(A \setminus \{p_i\}) + H(S).$$

This implies that $\mathrm{rank}_\Pi(A)$ is an integer if and only if $\mathrm{rank}_\Pi(A \setminus \{p_i\})$ is an integer, contradicting the choice of A as a minimal set whose rank is not an integer.

Case II: For every $p_i \in A$, the set $A \cup B \setminus \{p_i\}$ is in Γ. Let $A' \subseteq A$ be a minimal set such that $A' \cup B \in \Gamma$ and take a party $p_i \in A'$ (since $B \notin \Gamma$ such p_i exists). By (A.2), (A.4), and Claim 8.14,

$$0 \leq H(\{p_i\} | S, A \cup B \setminus \{p_i\}) \leq H(\{p_i\} | S, A' \cup B \setminus \{p_i\}) = 0. \tag{8.6}$$

Since $A \cup B \setminus \{p_i\} \in \Gamma$, $H(S|A \cup B \setminus \{p_i\}) = 0$ and by (A.3),(A.4), and (8.6),

$$\begin{aligned} 0 &\leq H(\{p_i\} | A \cup B \setminus \{p_i\}) \\ &\leq H(S, \{p_i\} | A \cup B \setminus \{p_i\}) \\ &= H(S|A \cup B \setminus \{p_i\}) + H(\{p_i\} | S, A \cup B \setminus \{p_i\}) = 0. \end{aligned} \tag{8.7}$$

By (A.3) and by Claim 8.15 (since B is a minimal set such that $A \cup B \in \Gamma$):

$$H(AB) = H(A) + H(B|A) = H(A) + |B| \cdot H(S), \tag{8.8}$$

and by (8.7), (A.3), and Claim 8.15 (since B is a minimal set such that $A \cup B \setminus \{p_i\} \in \Gamma$):

$$H(AB) = H(A \setminus \{p_i\}) + H(B|A \setminus \{p_i\}) + H(\{p_i\} | A \cup B \setminus \{p_i\}) = H(A \setminus \{p_i\}) + |B| \cdot H(S). \tag{8.9}$$

We deduce that, by (8.8) and (8.9), in this case $\mathrm{rank}_\Pi(A) = \mathrm{rank}_\Pi(A \setminus \{p_i\})$, contradicting the choice of A as a minimal set whose rank is not an integer.

□

We now prove that $\mathrm{rank}_\Pi : 2^{P \cup \{p_0\}} \to \mathbb{N}$ is a rank function of a matroid.

Claim 8.17. $\mathcal{M}_\Pi = \langle P \cup \{p_0\}, \mathrm{rank}_\Pi \rangle$ *is a matroid.*

Proof. The non-negativity, monotonicity, and sub-modularity follow from Theorem 7.3. By (A.2), $H(\{p_i\}) \leq H(S)$, thus boundedness follows by using (A.6):

$$\mathrm{rank}_\Pi(A) = \frac{H(A)}{H(S)} \leq \frac{\sum_{p_i \in A} H(\{p_i\})}{H(S)} \leq |A|.$$

□

We complete the proof by proving that Γ is the port of the matroid $\mathcal{M}_\Pi$.

Claim 8.18. *The access structure* Γ *is the port of the matroid* $\mathcal{M}_\Pi = \langle P \cup \{p_0\}, \mathrm{rank}_\Pi \rangle$ *at* p_0.

Proof. First, assume that A is a minimal authorized set in Γ, thus $H(S|A) = 0$ and

$$\mathrm{rank}_\Pi(A \cup \{p_0\}) = H(S, A)/H(S) = (H(A) + H(S|A))/H(S) = H(A)/H(S) = \mathrm{rank}_\Pi(A).$$

thus $A \cup \{p_0\}$ is a dependent set in the matroid. We still need to prove that $A \cup \{p_0\}$ is a *minimal* dependent set. By Claim 8.15 (with $B = A$ and $C = \emptyset$), $\mathrm{rank}_\Pi(A) = |A|$, i.e., A is independent. For every $A' \subsetneq A$, if $A' \cup \{p_0\}$ is dependent, then (since A' is independent)

$$\mathrm{rank}_\Pi(A') = \mathrm{rank}_\Pi(A' \cup \{p_0\}) = (H(A') + H(S|A'))/H(S) = \mathrm{rank}_\Pi(A') + (H(S|A'))/H(S),$$

i.e., $H(S|A') = 0$ contradicting the fact that A is a minimal authorized set.

Next assume that $A \cup \{p_0\}$ is a circuit in the matroid $\mathcal{M}_\Pi$. In particular,

$$|A| = \mathrm{rank}_\Pi(A) = \mathrm{rank}_\Pi(A \cup \{p_0\}) = (H(A) + H(S|A))/H(S) = |A| + (H(S|A))/H(S).$$

Thus, $H(S|A) = 0$ and A is authorized. If A is not a minimal authorized set, then it contains a minimal authorized set $A' \subsetneq A$ and by the first direction $A' \cup \{p_0\}$ is a circuit, contradicting the fact that $A \cup \{p_0\}$ is a circuit. □

To conclude, given the ideal access structure Γ, we defined a function rank_Π from an ideal scheme Π realizing Γ. We then proved that rank_Π is a rank function of a matroid, that is, it is an integral function satisfying the axioms of a matroid. Finally, we proved that Γ is the port of this matroid. □

Example 8.19. Consider the threshold access structure Γ_t, which consists of all subsets of parties of size at least t, and Shamir's scheme [163] (described in Section 2.2); this scheme realizes Γ_t and the scheme is ideal when the size of the domain of secrets is a prime-power greater than n. In this scheme, to share a secret s, the dealer randomly chooses a random polynomial $P(x)$ of degree $t-1$ such that $P(0) = s$, and the share of the ith party is $P(i)$. In Shamir's scheme, the shares of every set of t parties are uniformly distributed, i.e., when the secret is uniformly distributed, for every set $A \subset P \cup \{p_0\}$ of size at most t the entropy of S_A is $|A| \cdot H(S)$. On the other hand, since every t points determine a unique polynomial of degree $t-1$, in Shamir's scheme every t shares determine all other shares and the secret, i.e., for every set $A \subseteq P \cup \{p_0\}$ of size greater than t the entropy of S_A is $t \cdot H(S)$. Defining the rank function for Shamir's scheme as in the proof of Theorem 8.13, we get $\text{rank}_{\text{Shamir}}(A) = \min\{t, |A|\}$. This is the rank of the so-called uniform matroid $U_{t,n+1}$; the circuits of this matroid are all sets of size $t+1$. Indeed, Γ_t is the port of the matroid $U_{t,n+1}$ at p_0, as all its circuits containing p_0 are all sets of parties of size t.

8.4 Additional Results on Ideal Access Structures

Following Brickell and Davenport [50], many works have considered the characterization of ideal access structures. Alas, an exact characterization of ideal access structures is still an intriguing open problem. Seymour [162] has shown that the port of the Vámos matroid is not ideal (i.e., the necessary condition is not sufficient). Beimel et al. [26] showed that the port of the Vámos matroid is far from ideal, i.e., its information ratio is at least 1.1 (this constant was improved to 1.142 in a sequence of papers [136, 96, 82, 103]). Matúš [133] has shown that ideal secret-sharing schemes realizing an access structure Γ are closely related to the solutions of a system of generalized quasigroup equations to the matroid of the access structure Γ. Using this relation, Matúš has shown that ports of infinitely many matroids (including the non-Desargues matroids and matroids that contain as restrictions both the Fano and non-Fano matroids) do not have ideal schemes.

Simonis and Ashikhmin [167] considered the port of the Non-Pappus matroid. They constructed an ideal multilinear secret-sharing scheme realizing this access structure, where the secret and each share contain two field elements, and they proved (using known results about matroids) that there is no ideal linear secret-sharing realizing this access structure. This implies that the sufficient condition in Theorem 8.12 is stronger than the sufficient condition in Theorem 8.11. Other examples of ports of matroids that have ideal multilinear secret-sharing schemes and do not have ideal linear secret-sharing schemes were given in [148, 17, 33, 12]. Kaboli et al. [110] showed that there are ideal secret-sharing schemes that are not multilinear; however, this does not show that there is an access structure that has an ideal scheme that is not multilinear. It is an open question if there is an ideal access structure that does not have an ideal multilinear secret-sharing scheme.

Martí-Farré and Padró [128] showed that if an access structure is not a matroid port, then the information ratio of every secret-sharing scheme realizing it is at least 1.5 (compared to information ratio 1 of ideal schemes). The proof uses a forbidden minor characterization of matroid ports that was given by Seymour [161] and the fact that all these forbidden minors have information ratio at least 1.5. We note that such a gap does not exist for matroid ports that are not ideal, i.e., the port of the Vámos matroid has information ratio bigger than 1 [17] and smaller than 1.33 [128] and the port of the Fano-Non Fano matroid has information ratio 1 in the limit [135, 25] but is not ideal [133].

Beimel and Chor [18], Matúš [132], and Golic [99] have proved that an access structure Γ is ideal over a binary domain of secrets if and only if Γ is a port of matroid representable over $\mathbb{F}_2$. In addition, Beimel and Chor [18] and Matúš [132] have proved that an access structure Γ is ideal over a ternary domain of secrets if and only if Γ is a port of matroid representable over $\mathbb{F}_3$. That is, for a domain of secrets of size at most 3, the sufficient and necessary conditions of [50] are equivalent.

For many families of access structures, it was shown that the sufficient and necessary conditions collide, that is, an access structure in such a family is ideal if and only if the access structure is a port of a representable matroid. Such families include, for instance, the access structures on sets of four [168] and five [108] parties, the ones defined by graphs [44, 50, 53], those with three or four minimal qualified subsets [127], weighted threshold access structures [137, 146, 29], and bipartite access structures [146, 142, 143]. Ideal secret-sharing schemes for hierarchical access structures were studied in [165, 49, 172, 174]. Farrás et al. [83, 85] introduced integer polymatroids as a tool for characterizing ideal multi-partite secret-sharing schemes.

Chapter 9

Computational Secret Sharing

In this chapter, we define computational secret-sharing schemes and describe four computational secret-sharing schemes, showing that computational secret-sharing schemes can be more efficient than information-theoretic secret-sharing schemes.

9.1 Definition of Computational Secret-Sharing Schemes

We next define computational secret-sharing schemes, where the sharing and reconstruction are computed in polynomial time and a *polynomial-time adversary* controlling an unauthorized set of parties cannot distinguish between shares of one secret and shares of a different secret. This is in contrast to Definition 3.4 – the definition of secret-sharing with information-theoretic security – where we required the same indistinguishability for any *unbounded* adversary.

The input to a computational secret-sharing scheme also contains a security parameter 1^λ (in a unary representation) allowing to "measure" the required hardness of the cryptographic primitive used in the scheme (e.g., the length of the primes in the RSA encryption scheme). We require that an adversary whose running time is polynomial in the security parameters (and other parameters of the scheme) cannot break a computational secret-sharing scheme. When defining "efficiency" of a computational secret-sharing scheme, it is important to consider the way the access structure is represented, e.g., we can represent it as a monotone formula, a monotone circuit, or as a general (non-monotone) circuit. It will be convenient to represent an n-party access structure by its characteristic function $f : \{0,1\}^n \to \{0,1\}$ (see Definition 3.3).

Definition 9.1 (Representation model)**.** *A* representation model *is a polynomial time computable function $U : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$, where $U(P,x)$ is referred to as the value returned by a "program" P on an input x. We assume that each P specifies a number of parties n (i.e., if $x \notin \{0,1\}^n$, then $P(U,x) = 0$) and assume that $|P| \geq n$. We say that P represents the function $f : \{0,1\}^n \to \{0,1\}$ in the representation model U if $U(P,x) = f(x)$.*

We will usually consider universal representation models, which can represent every access structure (alas, most access structures will have an exponential-size representation). However, we will also use the

notation of representation to consider a (sequence) of specific access structures, e.g., to consider $n/2$-out-of-n threshold access structures, we represent the n-th threshold function by 1^n.

Definition 9.2 (Computational Secret Sharing)**.** *A computational secret-sharing scheme for a representation model U is a pair of algorithms $\langle \mathsf{Share}, \mathsf{Recon} \rangle$, where:*

- Share *is a randomized polynomial-time algorithm whose inputs are a security parameter 1^λ, a program P, and a secret $s \in \{0,1\}^*$. The output of* Share *is n shares $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$, where n denotes the number of parties specified by P.*

- Recon *is a deterministic polynomial-time reconstruction algorithm, whose inputs are a program P, an input $x \in \{0,1\}^n$, and shares $\langle \mathsf{sh}_i \rangle_{p_i \in I_x}$ of the parties in $I_x = \{p_i : x_i = 1\}$. The output of* Recon *is a secret s.*

The computational secret-sharing scheme $\langle \mathsf{Share}, \mathsf{Recon} \rangle$ realizes the representation model U if the following two requirements hold:

Correctness. *Algorithm* Recon efficiently *reconstructs the secret s using the shares of any authorized set of parties. That is, there exists a negligible function* $\mathrm{negl}(\lambda)$ *such that for every λ, program P, input x such that $U(P,x) = 1$, and secret s*

$$\textit{If } \langle \mathsf{sh}_i \rangle_{1 \le i \le n} \leftarrow \mathsf{Share}(1^\lambda, P, s), \textit{ then } \Pr[\mathsf{Recon}(P, x, \langle \mathsf{sh}_i \rangle_{i \in I_x}) = s] \ge 1 - \mathrm{negl}(\lambda),$$

where the probability is over the randomness of the the algorithm Share*.*

Computational Security. *Every non-uniform polynomial-time adversary controlling an unauthorized set I_x cannot deduce any information about the secret from the shares of I_x. Formally, we consider the following game between a non-uniform polynomial-time adversary and the dealer:*

- *The adversary, with input 1^λ, chooses a program P, an input $x \in \{0,1\}^n$ such that $U(P,x) = 0$ (where n is the number of parties specified by P), and two secrets s_0, s_1 such that $|s_0| = |s_1|$ and sends P, x, s_0, s_1 to the dealer.*
- *The dealer, which knows 1^λ, picks with uniform distribution a bit $b \in \{0,1\}$, computes*

$$\langle \mathsf{sh}_i \rangle_{1 \le i \le n} \leftarrow \mathsf{Share}(1^\lambda, P, s_b),$$

  and sends $\langle \mathsf{sh}_i \rangle_{p_i \in I_x}$ to the adversary.
- *The adversary outputs a bit b'.*

The adversary wins if $b = b'$. The secret-sharing scheme $\langle \mathsf{Share}, \mathsf{Recon} \rangle$ is computationally-secure if for every non-uniform polynomial-time adversary $\mathcal{A}$ there exists a negligible function $\mathrm{negl}_{\mathcal{A}}$ such that the probability that $\mathcal{A}$ wins in the above game is at most $1/2 + \mathrm{negl}_{\mathcal{A}}(\lambda)$.

Note that the share size in a computational secret-sharing scheme is polynomial in the security parameter, the size of the representation of the access structure, and the size of the secret. If the representation size is polynomial in the number of parties, the scheme has polynomial share size.

Remark 9.3. It suffices to construct a computational secret-sharing scheme where the size of the secret is $O(\lambda)$ as we next explain. Let $\langle \mathsf{Share}, \mathsf{Recon} \rangle$ be a secret sharing scheme in which the share size for a program P is $\text{size}_P(\lambda,|s|)$. Given a program P, and a long secret s, the dealer does the following:

- Generate a key k of size λ for a semantically-secure symmetric encryption scheme $\langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$.
- Share k using Share, i.e., computing $\langle \mathsf{sh}_i \rangle_{1 \leq i \leq n} \leftarrow \mathsf{Share}(1^\lambda, P, k)$.
- Compute $c \leftarrow \mathsf{Enc}(k, s)$.
- The share of party p_i is $\langle \mathsf{sh}_i, c \rangle$.[24]

The resulting share size for sharing an ℓ-bit secret is $O(\ell) + \text{size}_P(\lambda, \lambda)$.

9.2 Computational Threshold Secret Sharing

We have seen in Lemma 7.1 that in any information-theoretic secret-sharing scheme the size of the share of each party is at least the size of the secret. Krawczyk [117] showed a computational t-out-of-n secret-sharing scheme in which for large thresholds t the share size is much shorter than the size of the secret, i.e., share size $O(\ell/t + \lambda)$. For long secrets, this share size is almost optimal as the correctness requirement that every set of t parties can reconstruct the secret (without any security requirements) already implies that the share share size is at least ℓ/t.

Krawczyk's idea is to start with the construction of Remark 9.3; however, instead of giving the encryption c to each party, the dealer ensures that every set of t parties can recover c. This is done using Rabin's information dispersal scheme [152] (i.e., an MDS code), which ensures this property, i.e., it provides the correctness of a t-out-of-n secret-sharing scheme (without requiring any security). Rabin's scheme is similar to Shamir's scheme, i.e., the string given to each party is an evaluation of a polynomial $P(x) \stackrel{\text{def}}{=} \sum_{i=0}^{t-1} a_i x^i$ of degree $t-1$, where in Rabin's scheme the t coefficients of the polynomial are the message (in Shamir's scheme a_0 is the secret, while $a_1, \ldots, a_{t-1}$ are random elements). Rabin's information dispersal scheme is described in Figure 9.1. In Rabin's scheme, each set of t parties can recover the polynomial and therefore the message. The size of the string given to each party when storing a message of length $\ell > t \lceil \log(n) \rceil$ is $\lceil \ell/t \rceil$ (i.e., by working over the field $\mathbb{F}_{2^{\lceil \ell/t \rceil}}$).

Krawczyk's computational t-out-of-n secret-sharing scheme is described in Figure 9.2. The scheme uses a semantically-secure encryption scheme $\langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ (which is implied by one-way functions [104]). We assume that the length of its key is $O(\lambda)$, where the security parameter λ is at least $\log(n)$; this implies that the share size in Shamir's scheme used in Krawczyk's computational scheme is $O(\lambda)$. We further assume

[24] Alternatively, one can publish c in a public repository, and the share of p_i is only sh_i.

that the length of an encryption of a message of length ℓ is $O(\ell)$, and ℓ – the size of the secret – is at least $t \lceil \log(n) \rceil$. Thus, $|\mathsf{sh}_i^{\text{Shamir}}| = O(\lambda)$ and $|\mathsf{sh}_i^{\text{Rabin}}| = \lceil \ell / t \rceil$, and the size of the share of each party is $O(\lambda + \ell / t)$.

Theorem 9.4. *If one-way functions exist, then Krawczyk's secret-sharing scheme is a computational t-out-of-n secret-sharing scheme with share size $O(\lambda + \ell / t)$, where $\ell \geq t \log(n)$ is the length of the secrets.*

Rabin's Information Dispersal Scheme

The message: t elements $a_0, \ldots, a_{t-1} \in \mathbb{F}_q$, where $q \geq n$ is a prime power.
The scheme:

- Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be n distinct non-zero elements known to all parties.
- The message defines a polynomial $P(x) = \sum_{i=0}^{t-1} a_i x^i$.
- The string given to p_j is $\mathsf{sh}_j \leftarrow P(\alpha_j)$.

Figure 9.1: Rabin's information dispersal scheme, where every t parties can recover the message.

Krawczyk's Computational t-out-of-n Scheme

The secret: a string $s \in \{0,1\}^{\ell}$, where $\ell \geq t \lceil \log(n) \rceil$.
The scheme:

- Generate a key $k \leftarrow \mathsf{Gen}(1^{\lambda})$ and Compute $c \leftarrow \mathsf{Enc}(k, s)$.
- Share the key k using Shamir's t-out-of-n secret-sharing scheme; let $\mathsf{sh}_1^{\text{Shamir}}, \ldots, \mathsf{sh}_n^{\text{Shamir}}$ be the generated shares.
- Encode c using Rabin's information dispersal scheme; let $\mathsf{sh}_1^{\text{Rabin}}, \ldots, \mathsf{sh}_n^{\text{Rabin}}$ be the generated strings.
- **Share of party p_i:** $\mathsf{sh}_i^{\text{Shamir}}, \mathsf{sh}_i^{\text{Rabin}}$.

Figure 9.2: Krawczyk's computational t-out-of-n secret-sharing scheme, using a semantically-secure encryption scheme $\langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$.

9.3 Computational Secret Sharing for Monotone Circuits

In Section 4.3 we described an information-theoretic secret-sharing scheme for monotone formulas of Benaloh and Leichter [36] (abbreviated the BL secret-sharing scheme). Can one extend this result to the more

powerful computational models of monotone circuits? An unpublished result of Yao [186] showed that there is a computational secret-sharing scheme for monotone circuits; a proof of the security of the scheme appeared in [180]. In this monograph, we will describe this result. We consider monotone circuits with AND and OR gates with unbounded fan-in. We refer the reader to Appendix A.1 or [109] for a reminder of the definition of monotone circuits.

To describe Yao's secret-sharing scheme, we first recall the BL secret-sharing scheme, described in Section 4.3. The scheme shares a secret s using a monotone formula F; it starts in the output gate of the formula, and in each node of the formula, the scheme recursively shares an appropriate secret for each child of the node. The total size of the shares in the BL secret-sharing scheme is the number of leaves in the formula. We can try to apply the same method for a monotone circuit. The problem is that if a node has fan-out greater than 1, it gets (from the recursion) a few values, and the dealer needs to share each value recursively. This can result in share size exponential in the number of nodes in the circuit. To overcome this problem, Yao suggested using encryption. For each node v in the circuit, the dealer chooses a key k_v for a semantically-secure symmetric encryption scheme $\langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$, encrypts the values given to the node v using the key k_v, and recursively shares the key k_v. The encryptions generated for each node are given to all parties. As only the key is shared recursively, the number of elements given to each node are its fan-out, i.e., the share size of each party is the number of edges in the circuit, which is at most quadratic in the size of the circuit. We formally describe the sharing algorithm of Yao's secret-sharing scheme in Figure 9.3.

We describe the reconstruction algorithm of Yao's secret-sharing schemes in Figure 9.4. Given the circuit C, whose nodes in a lexicographic order are $G_1, \ldots, G_g$, let $C_i(x)$ be the Boolean function computed by the node G_i on input $x \in \{0,1\}^n$. Fix an input x such that $f(x) = 1$. The reconstruction algorithm proceeds according to this lexicographic order, computes k_i for every node such that $C_i(x) = 1$ as explained below, and uses this key to obtain string_i. For leaves, the encryption key is part of the shares of I_x. For an OR gate, for at least one of its children j it must be that $C_j(x) = 1$, thus the reconstruction algorithm has already computed string_j and k_i is part of it. For an AND gate, for all its children j it must be that $C_j(x) = 1$, thus the reconstruction algorithm has already computed string_j for every child, and it can reconstruct k_i from these strings. Since $f(x) = 1$, the circuit C returns 1 on x, i.e., $C(x) = C_g(x) = 1$, thus, algorithm Yao-Reconstruct computes $\mathrm{string}_g = s$, i.e., reconstructs the secret. In algorithm Yao-Reconstruct, each string_i is computed at most once; for every node the complexity of computing the string is at most the fan-in of the node. Hence, the complexity of the reconstruction is linear in the number of edges in C. The correctness of the reconstruction algorithm follows by induction.

The security for an input x such that $f(x) = 0$ follows from the same arguments as in the BL secret-sharing scheme, i.e., an adversary controlling the parties in I_x cannot learn in polynomial time any information of the keys and strings of nodes G_i such that $C_i(x) = 0$. Intuitively, this follows by an inductive argument. For example, for an AND gate G_i such that $C_i(x) = 0$, there exists at least one incoming edge (G_j, G_i) such that $C_j(x) = 0$, thus, by the induction hypothesis, the adversary cannot learn in polynomial time any information about string_j, i.e., it cannot learn in polynomial-time any information of at least one of the shares in the g_i-out-of-g_i secret-sharing scheme of k_i; hence, the adversary cannot learn in polynomial-time

Yao's Computational Secret-Sharing Scheme

The secret: a string $s \in \{0,1\}^{\ell}$ for some $\ell \in \mathbb{N}$.
The security parameter: 1^{λ}.
The circuit: a monotone Boolean circuit C representing an access structure f; let $G_1, G_2, \ldots, G_g$ be the nodes of the circuit C sorted according to a topological order (that is, G_g is the root and G_j is the leaf labeled by x_j for $1 \leq j \leq n$).
The scheme:

1. $k_i \leftarrow \mathsf{Gen}(1^{\lambda})$ for each $1 \leq i \leq g$.
2. $\text{string}_g \leftarrow s$ and $\text{string}_i \leftarrow \varepsilon$ for each $1 \leq i \leq g-1$.
3. For $i = g$ downto 1:
   (a) $e_i \leftarrow \mathsf{Enc}(1^{\lambda}, k_i, \text{string}_i)$.
   (b) If $i > n$ do:
   - Let $G_{j_1}, \ldots, G_{j_{g_i}}$ be the sources of the incoming edges into the node G_i, where $g_i \geq 1$ and $j_1 < \cdots < j_{g_i} < i$.
   - If G_i is an OR gate, then $\text{string}_{j_\alpha} \leftarrow \text{string}_{j_\alpha}, k_i$ for $1 \leq \alpha \leq g_i$ (i.e., k_i is concatenated to string_i).
   - If G_i is an AND gate, then
     - Share k_i using a g_i-out-of-g_i secret-sharing scheme; let $k_{i,1}, \ldots, k_{i,g_i}$ be the shares of k_i (i.e., $k_i = \bigoplus_{\alpha=1}^{g_i} k_{i,\alpha}$).
     - $\text{string}_{j_\alpha} \leftarrow \text{string}_{j_\alpha}, k_{i,\alpha}$ for $1 \leq \alpha \leq g_i$.

   (c) **Share of party p_j:** The key k_j and the encryptions $e_1, \ldots, e_g$.

Figure 9.3: The sharing algorithm in Yao's secret-sharing scheme for a monotone circuit C.

Algorithm Yao-Reconstruct

The circuit: A monotone Boolean circuit C representing an access structure Γ. Let $G_1, G_2, \ldots, G_g$ be the nodes of the circuit C sorted according to a topological order and let C_i be the sub-circuit of C whose output is the output of G_i.
The reconstructing set: A set $I_x = \left\{p_i : x_i = 1\right\}$ for $x \in \{0,1\}^n$ such that $C(x) = 1$.
The reconstructing algorithm:

1. For $i = 1$ to n do (where G_i is a leaf labeled by x_i):
   (a) If $C_i(x) = 1$ (that is, $x_i = 1$, i.e., $p_i \in I_x$ and the shares of I_x contain k_i), then compute $\text{string}_i \leftarrow \mathsf{Dec}(1^\lambda, k_i, e_i)$.
2. For $i = n+1$ to g do:
   (a) If G_i is an OR gate and $C_i(x) = 1$ (that is, there exists an incoming edge (G_j, G_i) such that $C_j(x) = 1$ and $j < i$), then take k_i from string_j and compute $\text{string}_i \leftarrow \mathsf{Dec}(1^\lambda, k_i, e_i)$.
   (b) If G_i is an AND gate and $C_i(x) = 1$ (that is, for every incoming edge (G_j, G_i) it must be that $C_j(x) = 1$), then for the incoming edges from $G_{j_1}, \ldots, G_{j_{g_i}}$ to G_i, where $j_1 < \cdots < j_{g_i} < i$:
      - For every $1 \leq \alpha \leq g_i$ do:
        - Take $k_{i,\alpha}$ from string_j.
      - Compute $k_i \leftarrow \bigoplus_{\alpha=1}^{g_i} k_{i,\alpha}$ and $\text{string}_i \leftarrow \mathsf{Dec}(1^\lambda, k_i, e_i)$.
3. If $C(x) = 1$, then output string_g.

Figure 9.4: The reconstruction algorithm in Yao's secret-sharing scheme, which computes string_i for every node G_i such that $C_i(x) = 1$.

any information on k_i.

We formalize this idea for the computational setting in Claim 9.5, where we use the so-called hybrid argument.

Claim 9.5. *Yao's secret-sharing scheme is secure.*

Proof. Let $\mathcal{A}$ be a non-uniform polynomial time adversary trying to break Yao's secret-sharing scheme. Fix the security parameter λ. Consider the choices of the adversary $\mathcal{A}$ with the security parameter 1^λ: the circuit C, the input x such $C(x) = 0$, and the secrets s_0, s_1. Let n be the number of inputs of C. We consider 4 distributions. The first two distributions, denoted $\mathcal{H}_0^b$ for $b \in \{0,1\}$, are encryptions $e_1, \ldots, e_g$ and keys $\langle k_i \rangle_{p_i \in I_x}$, where $e_1 = \mathsf{Enc}(1^\lambda, k_1, \mathrm{string}_i), \ldots, e_g = \mathsf{Enc}(1^\lambda, k_g, \mathrm{string}_g)$ for $\mathrm{string}_1, \ldots, \mathrm{string}_g$ generated by Yao's secret-sharing scheme for the secret s_b (that is, $\mathrm{string}_g = s_b$). The last two distributions, denoted $\mathcal{H}_g^b$ for $b \in \{0,1\}$, are the encryptions $e'_1, \ldots, e'_g$ and keys $\langle k_i \rangle_{p_i \in I_x}$, computed from the strings $\mathrm{string}_1, \ldots, \mathrm{string}_g$ generated by Yao's secret-sharing scheme for the secret s_b as follows:

- If $C_i(x) = 1$, then $\mathrm{string}'_i \leftarrow \mathrm{string}_i$.
- If $C_i(x) = 0$, then $\mathrm{string}'_i \leftarrow 0^{|\mathrm{string}_i|}$,

and $e'_i = \mathsf{Enc}(1^\lambda, k_i, \mathrm{string}'_i)$. In other words, $e'_1, \ldots, e'_g$ contains encryptions of all strings that the parties in I_x know and the other encryptions are the encryptions of the all-zero strings (of appropriate length). As $C(x) = C_g(x) = 0$, the string string'_g is the all-zero string, hence $\mathcal{H}_g^0$ and $\mathcal{H}_g^1$ are actually the same distribution. We will show that for every adversary $\mathcal{A}$ there is a negligible function negl such that for every $b \in \{0,1\}$

$$\left| \Pr_{\mathbf{h} \in_R \mathcal{H}_0^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \Pr_{\mathbf{h} \in_R \mathcal{H}_g^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] \right| \leq \mathrm{negl}(\lambda). \tag{9.1}$$

We first show that (9.1) implies that $\mathcal{A}$ has negligible advantage on guessing b. The adversary $\mathcal{A}$ wins the security game of Definition 9.2 if it gets a sample from $\mathcal{H}_0^1$ (i.e., sharing of s_1) and answers 1 or gets a sample from $\mathcal{H}_0^0$ (i.e., sharing of s_0) and answers 0, i.e.,

$$\begin{aligned}
&\Pr[\mathcal{A} \text{ wins the security game of Definition 9.2 }] \\
&\quad = 0.5 \cdot \Pr_{\mathbf{h} \in_R \mathcal{H}_0^1}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] + 0.5 \cdot \Pr_{\mathbf{h} \in_R \mathcal{H}_0^0}[\mathcal{A}(1^\lambda, \mathbf{h}) = 0] \\
&\quad = 0.5 \cdot \Pr_{\mathbf{h} \in_R \mathcal{H}_0^1}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] + 0.5 \cdot (1 - \Pr_{\mathbf{h} \in_R \mathcal{H}_0^0}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1]) \\
&\quad \leq 0.5 + 0.5 \cdot \left(\Pr_{\mathbf{h} \in_R \mathcal{H}_g^1}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] + \mathrm{negl}(\lambda) \right) - 0.5 \cdot \left(\Pr_{\mathbf{h} \in_R \mathcal{H}_g^0}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \mathrm{negl}(\lambda) \right) \\
&\quad = 0.5 + 2\,\mathrm{negl}(\lambda),
\end{aligned}$$

where the inequality follows from (9.1) and the last equality follows from the fact that $\mathcal{H}_g^0$ and $\mathcal{H}_g^1$ are identical. Thus, to complete the proof of security we need to prove (9.1). For this, we define "hybrid" distributions

$\mathcal{H}_1^b, \ldots, \mathcal{H}_{g-1}^b$, where the distribution $\mathcal{H}_i^b$ is the encryptions $e'_1, \ldots, e'_i, e_{i+1}, \ldots, e_g$ and keys $\langle k_i \rangle_{p_i \in I_x}$, i.e., the first i encryptions are generated as in $\mathcal{H}_g^b$ and the last $g-i$ encryptions are generated as in $\mathcal{H}_0^b$. We will show that the semantic-security of $\langle \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec} \rangle$ implies that for every non-uniform polynomial-time algorithm $\mathcal{A}$ there exist a negligible function $\mathrm{negl}_b(\lambda)$ such that:

$$\left| \Pr_{\mathbf{h} \in_R \mathcal{H}_i^b} [\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \Pr_{\mathbf{h} \in_R \mathcal{H}_{i-1}^b} [\mathcal{A}(1^\lambda, \mathbf{h}) = 1] \right| \leq \mathrm{negl}_b(\lambda), \tag{9.2}$$

where the notation $\mathbf{h} \in_R \mathcal{H}_i^b$ denotes a vector of encryptions and keys generated as in $\mathcal{H}_i^b$. Note that $\mathcal{H}_{i-1}^b$ and $\mathcal{H}_i^b$ only differ on the i-th encryption, which is an encryption of string'_i in $\mathcal{H}_i^b$ and the encryption of string_i in $\mathcal{H}_{i-1}^b$. If $C_i(x) = 1$, the distributions $\mathcal{H}_i^b, \mathcal{H}_{i-1}^b$ are the same and (9.2) is trivial in this case.

To prove (9.2) when $C_i(x) = 0$, we construct an adversary $\mathcal{B}$ from $\mathcal{A}$ that tries to break the encryption scheme. We recall that in the security game for the encryption, $\mathcal{B}$ chooses two messages m_0, m_1, sends them to the encrypter, gets from the encrypter $c \leftarrow \mathsf{Enc}(1^\lambda, k, m_d)$ for a uniformly distributed $d \in \{0,1\}$ and a key $k \leftarrow \mathsf{Gen}(1^\lambda)$, and tries to guess d. The adversary $\mathcal{B}$ on input 1^λ and i does the following:

- Generates n keys $k_1, \ldots, k_g$.
- Generates strings $\mathrm{string}'_1, \ldots, \mathrm{string}'_{i-1}, \mathrm{string}_i, \ldots, \mathrm{string}_g$ as in $\mathcal{H}_{i-1}^b$ and $\mathrm{string}'_i = 0^{|\mathrm{string}_i|}$ and computes $e'_j = \mathsf{Enc}(1^\lambda, k_j, \mathrm{string}'_j)$ for $1 \leq j \leq i-1$ and $e_j = \mathsf{Enc}(1^\lambda, k_j, \mathrm{string}_j)$ for $i+1 \leq j \leq g$.
- Sends $m_0 \leftarrow \mathrm{string}_i$ and $m_1 \leftarrow \mathrm{string}'_i = 0^{|\mathrm{string}_i|}$ to the encrypter and gets an encryption $e = \mathsf{Enc}(1^\lambda, k, m_d)$.
- Sends $e'_1, \ldots, e'_{i-1}, e, e_{i+1}, \ldots, e_n$ and $\left\langle k_j \right\rangle_{p_j \in I_x}$ to $\mathcal{A}$, gets a bit d', and outputs d'.

First, note that k_i is not used to encode any string. Second, we claim that $\mathrm{string}'_1, \ldots, \mathrm{string}'_{i-1}, \mathrm{string}_{i+1}, \ldots, \mathrm{string}_g$ and keys $\left\langle k_j \right\rangle_{p_j \in I_x}$ are independent of the choice of k_i. There are three cases to consider.

$1 \leq i \leq n$. The key k_i of a leaf does not appear in any string. Furthermore, as $x_i = 0$, party p_i is not in I_x and k_i is not given to the parties in I_x.

G_i is an OR gate. For every incoming edge (G_j, G_i) it must be that $j < i$ (since the nodes are ordered in a lexicographic order) and $C_j(x) = 0$ (since $C_i(x) = 0$ and G_i is an OR gate), thus, string'_j is the all-zero string and k_i does not appear in any string $\mathrm{string}'_1, \ldots, \mathrm{string}'_{i-1}$. Furthermore, k_i is independent of the keys of $\left\langle k_j \right\rangle_{p_j \in I_x} \subseteq \left\{k_1, \ldots, k_n\right\}$ since $i > n$.

G_i is an AND gate. There is an incoming edge (G_j, G_i) such that $j < i$ and $C_j(x) = 0$ (since $C_i(x) = 0$ and G_i is an AND gate), thus, string'_j is the all-zero string and at least one of the shares of k_i in the g_i-out-of-g_i secret-sharing scheme does not appear the strings $\mathrm{string}'_1, \ldots, \mathrm{string}'_{i-1}$. By the security of the g_i-out-of-g_i secret-sharing scheme, the strings $\mathrm{string}'_1, \ldots, \mathrm{string}'_{i-1}$ are independent of k_i.

We use the above fact to complete the proof of the security. If the encrypter chooses $d = 1$, then $e = \mathsf{Enc}(1^\lambda, k, 0^{|\text{string}_i|})$, i.e., $e'_1, \ldots, e'_{i-1}, e, e_{i+1}, \ldots, e_n$ and $\left\langle k_j \right\rangle_{p_j \in I_x}$ are generated as in $\mathcal{H}_i^b$ with the keys $\left\langle k_j \right\rangle_{j \neq i}$ and k, and

$$\Pr\left[\mathcal{B}\left(1^\lambda, i, e = \mathsf{Enc}(1^\lambda, k, m_1)\right) = 1\right] = \Pr_{\mathbf{h} \in_R \mathcal{H}_i^b}\left[\mathcal{A}(1^\lambda, \mathbf{h}) = 1\right].$$

Similarly, if the encrypter chooses $d = 0$, then $e = \mathsf{Enc}(1^\lambda, k, \text{string}_i)$, i.e., $e'_1, \ldots, e'_{i-1}, e, e_{i+1}, \ldots, e_n$ and $\left\langle k_j \right\rangle_{p_j \in I_x}$ are generated as in $\mathcal{H}_{i-1}^b$ with the keys $\left\langle k_j \right\rangle_{j \neq i}$ and k, and

$$\Pr\left[\mathcal{B}\left(1^\lambda, i, e = \mathsf{Enc}(1^\lambda, k, m_0)\right) = 1\right] = \Pr_{\mathbf{h} \in_R \mathcal{H}_{i-1}^b}\left[\mathcal{A}(1^\lambda, \mathbf{h}) = 1\right].$$

By the security of the encryption scheme, there exists a negligible function $\text{negl}_b(\lambda)$ such that for every i

$$\left|\Pr[\mathcal{B}(1^\lambda, i, e = \mathsf{Enc}(1^\lambda, k, m_1)) = 1] - \Pr[\mathcal{B}(1^\lambda, i, e = \mathsf{Enc}(1^\lambda, k, m_0)) = 1]\right| \leq \text{negl}_b(\lambda)$$

(where we consider a non-uniform adversary $\mathcal{B}$ that gets a worst i_λ for every λ).

Thus, (9.2) follows. We conclude that

$$\begin{aligned}
&\left|\Pr_{\mathbf{h} \in_R \mathcal{H}_g^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \Pr_{\mathbf{h} \in_R \mathcal{H}_0^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1]\right| \\
&\qquad = \left|\sum_{i=1}^{g} \Pr_{\mathbf{h} \in_R \mathcal{H}_i^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \sum_{i=1}^{g} \Pr_{\mathbf{h} \in_R \mathcal{H}_{i-1}^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1]\right| \\
&\qquad \leq \sum_{i=1}^{g} \left|\Pr_{\mathbf{h} \in_R \mathcal{H}_i^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1] - \Pr_{\mathbf{h} \in_R \mathcal{H}_{i-1}^b}[\mathcal{A}(1^\lambda, \mathbf{h}) = 1]\right| \\
&\qquad \leq \sum_{i=1}^{g} \text{negl}_b(\lambda).
\end{aligned}$$

As the adversary $\mathcal{A}$ runs in time polynomial in λ, the number of parties n is bounded by a polynomial in λ, hence g is bounded by a polynomial in λ. As the sum of polynomially many negligible functions is negligible $\sum_{i=1}^{g} \text{negl}_b(\lambda)$ is negligible as claimed. □

Yao's secret-sharing scheme uses a semantically-secure encryption scheme, which, by [104], exists assuming that one-way functions exist.

Theorem 9.6. *If one-way functions exist, then Yao's secret-sharing scheme is a computational secret-sharing for monotone circuits with share size $O(\ell + |C| \cdot \lambda)$, where λ is the security parameter, ℓ is the length of the secret, and $|C|$ is the number of wires in the circuit C.*

Remark 9.7. In Yao's secret-sharing scheme, the share of each party p_i contains its key k_i and the encryptions of the strings for each vertex in the circuit. If we allow public information (e.g., information stored on some public repository), then the encryptions can be published; the share size of each party reduces to $O(\lambda)$ and the size of the public information is $O(\ell + |C| \cdot \lambda)$.

Remark 9.8. In the description of Yao's scheme, we represented an access structure by a monotone circuit with AND and OR gates. We can generalize the above protocol by adding other gates to the formula (e.g., threshold gates); we require that each such gate fan-in g can be realized by an information-theoretic secret-sharing scheme with $\text{poly}(g)$ share size. To share s secret, we replace Item 3b in Yao's scheme (described in Figure 9.3) by the following loop

- If $i > n$ do:
  - Share k_i using a secret-sharing scheme with perfect security; let $k_{i,1}, \ldots, k_{i,g_i}$ be the shares of k_i.
  - $\text{string}_{j_\alpha} \leftarrow \text{string}_{j_\alpha}, k_{i,\alpha}$ for $1 \leq \alpha \leq g_i$.

Let size_i be the total share size for realizing gate G_i in the circuit with a one-bit secret. The share size of the generalized Yao's scheme is $O(\ell + (\sum_{i=n+1}^{g} \text{size}_i) \cdot \lambda)$.

9.4 Computational Secret Sharing for Circuits

Komargodski et al. [116] proved that under strong cryptographic assumptions, all access structures represented by general (possibly non-monotone) circuits can be realized by a computational secret-sharing scheme (as the circuit represents a monotone access structure, the circuit computes a monotone function). This is a much stronger representation than monotone circuits, i.e., for some monotone languages monotone circuits can be exponentially bigger than non-monotone circuits [171]. We start with a motivating example.

Definition 9.9. *Let $G_n = (V, E)$ be a complete undirected graph with m vertices and $n = \binom{m}{2}$ edges, where m is even. A perfect matching in the graph $G_n = (V, E)$ is a set $M \subseteq E$ of size $|V|/2$ such that each $v \in V$ appears in exactly one edge in M (i.e., each vertex is matched to its unique neighbor). The matching access structure, denoted Γ_{match}, is the access structure whose parties are edges of the complete undirected graph G_n and its authorized sets are subsets of the edges containing a perfect matching in G_n.*

Are there information-theoretic or computational secret-sharing schemes realizing Γ_{match} with polynomial-size shares? Matching does not have polynomial size monotone circuits [155] or polynomial size monotone span programs [150].[25] Thus, the constructions that we have seen so far do not provide efficient schemes for Γ_{match}. As perfect matching can be computed by a polynomial-size circuit, the construction we present in this section will realize Γ_{match} with computational security and polynomial share size.

Komargodski et al. [116] constructed computational secret-sharing schemes for circuits, which we call the KNY secret-sharing scheme. The KNY scheme uses a primitive called a witness encryption scheme, a modern primitive introduced by Garg et al. [91]. Originally, witness encryption schemes were constructed from indistinguishability obfuscation (iO) and multilinear maps [91]; however, recent works [60, 13, 177, 179] have given direct constructions of witness encryption that are significantly more efficient than existing

[25] In [150], a lower bound of $n^{\log(n)}$ is proved for the directed s-t-connectivity access structure. As directed s-t-connectivity can be reduced to matching via a projection reduction, the same lower bound applies to Γ_{match}.

constructions of iO. In a witness encryption scheme for a language L in NP, a message M is encrypted with an input x. If $x \in L$ and a witness w for $x \in L$ is given, then the ciphertext can be efficiently decrypted. On the other hand, if $x \notin L$, then no polynomial-time adversary can learn any information on the message from the ciphertext. We quote the definition from [91].

Definition 9.10 (Witness Encryption [91])**.** *Let L be a language in n* NP *with corresponding witness relation R, that is, there is some $c \in \mathbb{N}$ such that $L = \left\{x : \exists_w |w| \leq |x|^c \wedge (x, w) \in R\right\}$. A witness encryption scheme with message space $\{0,1\}$ for L consists of the following two polynomial-time algorithms:*

Encryption. *The algorithm $\mathsf{WE.Enc}_L(1^\lambda, x, M)$ takes as input a security parameter 1^λ, a string x, and a message $M \in \{0,1\}$, and outputs a ciphertext* CT.

Decryption. *The algorithm $\mathsf{WE.Dec}_L(\mathrm{CT}, w)$ takes as input a ciphertext* CT *and a string w, and outputs a message M or the symbol $\perp$.*

These algorithms satisfy the following two conditions:

Correctness. *There exists a negligible function* $\mathrm{negl}(\lambda)$ *such that for any security parameter λ, message $M \in \{0,1\}$, input $x \in L$, and witness w such that $\langle x, w\rangle \in R$,*

$$\Pr\left[\mathsf{WE.Dec}_L(\mathsf{WE.Enc}_L(1^\lambda, x, M), w) = M\right] \geq 1 - \mathrm{negl}(\lambda).$$

Soundness Security. *For any non-uniform polynomial-time adversary $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\lambda)$ *such that for any $x \notin L$, we have:*

$$\Pr[\mathcal{A}(\mathsf{WE.Enc}_L(1^\lambda, x, 0)) = 1] - \Pr[\mathcal{A}(\mathsf{WE.Enc}_L(1^\lambda, x, 1)) = 1] < \mathrm{negl}(\lambda).$$

We will need the following observation.

Observation 9.11. *If there exists a witness encryption scheme for some* NP*-complete language (under Levin's reductions), then for every language in* NP *there is a witness encryption scheme. We denote such a witness encryption scheme as a witness encryption for* NP*.*

Historical Notes. The result of Komargodski et al. [116] is stronger than we stated it and their scheme realizes access structures represented by non-deterministic circuits (where the efficient reconstruction algorithm is also given a witness for the fact that the circuit accepts the input); their result is phrased as a secret-sharing scheme for every monotone language in NP. This strengthens a result of Garg et al. [91] that showed that a witness encryption scheme for NP implies a computational secret-sharing scheme for a specific NP-complete monotone language. Komargodski et al. also observed that if an NP-complete monotone language has a computational secret-sharing scheme, then there exists a witness encryption for NP. For simplicity of the presentation, we will not discuss secret-sharing for non-deterministic circuits.

The KNY secret-sharing scheme also uses a non-interactive commitment scheme with perfect binding. A commitment scheme is a digital analogue of envelopes, where a committer holds a message M and computes a commitment $\mathsf{Commit}(1^\lambda, M; r)$ with a uniformly chosen random string r. To open the commitment, the committer reveals r. A commitment scheme should satisfy hiding – a non-uniform polynomial-time adversary cannot learn in polynomial time any information on M from $\mathsf{Commit}(1^\lambda, M; r)$ – and binding – the committer can open a commitment only to one value. A formal definition is given in Appendix A.2.

The key idea in the KNY scheme is using the following language for the witness encryption scheme:

$$\mathrm{Com} = \left\{ \left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle : \begin{array}{l} C \text{ is a circuit with } n \text{ inputs for some } n \in \mathbb{N}, \\ \exists_{x \in \{0,1\}^n, r_1, \ldots, r_n} C(x) = 1 \wedge \bigwedge_{1 \le i \le n} \left(x_i = 1 \rightarrow c_i = \mathsf{Commit}(1^\lambda, 1; r_i) \right) \end{array} \right\}.$$

Notice that if L is a language in P, then Com is in NP, where a witness for $\left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle \in \mathrm{Com}$ is $x \in \{0,1\}^n$ such that $C(x) = 1$ and $\langle r_i \rangle_{x_i = 1}$ such that if $x_i = 1$ then $c_i = \mathsf{Commit}(1^\lambda, 1; r_i)$; that is, the witness is the input x and an opening of the commitment for every bit that is 1 in x.

The Sharing Algorithm of the KNY Secret-Sharing Scheme

The secret: a bit $s \in \{0,1\}$.
The circuit: A circuit C with n inputs representing a monotone access structure f.
The security parameter: 1^λ.
The scheme:

- Choose with uniform distribution n random strings $r_1, \ldots, r_n$ for Commit.
- Let $c_i \leftarrow \mathsf{Commit}(1^\lambda, 1; r_i)$ for $1 \le i \le n$.
- Let $\mathrm{CT} \leftarrow \mathsf{WE.Enc}_{\mathrm{Com}}(1^\lambda, \left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle, s)$.
- **Share of party p_i:** CT, r_i.

Figure 9.5: The sharing algorithm of the KNY computational secret-sharing for circuits.

The sharing of the KNY secret-sharing scheme is described in Figure 9.5; for simplicity we assume that the secret is a bit (to share a longer secret we can share each bit independently). Informally, to share a secret s among n parties, the dealer commits to 1 for every $1 \le i \le n$, that is, computes $c_i \leftarrow \mathsf{Commit}(1^\lambda, 1; r_i)$, and encrypts s with a witness encryption scheme for Com with the word $\left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle$, let CT be the encryption. The share of p_i is CT, c_i.

We next explain how to reconstruct s from the shares of I_x for an input such that $f(x) = 1$. Recall that $C(x) = f(x) = 1$ and the parties in $I_x = \left\{ i : x_i = 1 \right\}$ hold the shares $\langle \mathrm{CT}, r_i \rangle_{i : x_i = 1}$, where r_i is an opening of c_i. As $x, \langle r_i \rangle_{i : x_i = 1}$ is a witness that $\left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle \in \mathrm{Com}$, the parties in I_x can decrypt CT and reconstruct s. The reconstruction algorithm of the KNY secret-sharing scheme is formally described in Figure 9.6.

The Reconstruction Algorithm of the KNY Secret-Sharing Scheme

The shares: $\langle \mathrm{CT}, r_i \rangle_{p_i \in I_x}$ for an input x such that $f(x) = 1$.
The circuit: A circuit C with n inputs representing a monotone access structure f.
The scheme:

- Let $s \leftarrow \mathsf{WE.Dec}_{\mathrm{Com}}\left(\mathrm{CT}, \left\langle x, \langle c_i \rangle_{i : p_i \in I_x} \right\rangle\right)$.
- **Output:** s.

Figure 9.6: The reconstruction algorithm of the KNY computational secret-sharing for circuits.

We next present an informal argument that the computational security of the KNY scheme.[26] Note that an adversary holds shares of an unauthorized set and gets an encryption CT for a word $\mathbf{c} = \left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle \in$ Com (assuming that the access structure is non-empty, i.e., $C(1^n) = 1$); however, the adversary does not know a witness for $\left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle \in$ Com. Thus, the security of the witness encryption scheme does not directly imply that the adversary cannot learn information on the secret. We will show that the security of the commitment scheme together with the security of the witness encryption scheme imply the security of the KNY secret-sharing scheme. Specifically, we will consider two vectors of commitments for an input x such that $C(x) = 0$:

$$\mathbf{c} = \langle c_1, \ldots, c_n \rangle, \text{ where } c_i \leftarrow \mathsf{Commit}(1^\lambda, 1; r_i) \text{ for every } 1 \leq i \leq n.$$

and

$$\mathbf{c}' = \left\langle c'_1, \ldots, c'_n \right\rangle, \text{ where } c'_i \leftarrow \begin{cases} \mathsf{Commit}(1^\lambda, 1; r'_i) & \text{if } x_i = 1, \\ \mathsf{Commit}(1^\lambda, 0; r'_i) & \text{if } x_i = 0. \end{cases}$$

By the perfect binding of Commit, the vector of commitments $\mathbf{c}'$ has a unique opening. As $C(x) = 0$ and C computes a monotone function, $\left\langle 1^\lambda, C, c'_1, \ldots, c'_n \right\rangle \notin$ Com. By the security of the witness encryption scheme an adversary cannot learn information on s from $\mathrm{CT}' \leftarrow \mathsf{WE.Enc}_{\mathrm{Com}}(1^\lambda, \mathbf{c}', s)$, and $\langle r_i \rangle_{x_i = 1}$. By the hiding property of Commit (using a hybrid argument), an adversary cannot distinguish with non-negligible probability between $\mathbf{c}$ and $\mathbf{c}'$ even when it holds $\langle r_i \rangle_{x_i = 1}$. Thus, an adversary cannot distinguish between $\mathrm{CT} \leftarrow \mathsf{WE.Enc}_{\mathrm{Com}}(1^\lambda, \left\langle 1^\lambda, C, c_1 \ldots, c_n \right\rangle, s)$ and $\mathrm{CT}' \leftarrow \mathsf{WE.Enc}_{\mathrm{Com}}(1^\lambda, \left\langle 1^\lambda, C, c'_1 \ldots, c'_n \right\rangle, s)$ (even if it knows $\langle r_i \rangle_{x_i = 1}$). It follows that a non-uniform polynomial-time adversary cannot learn information on s from $\mathrm{CT} \leftarrow \mathsf{WE.Enc}(1^\lambda, \left\langle 1^\lambda, C, c_1, \ldots, c_n \right\rangle, s), \langle r_i \rangle_{x_i = 1}$.

The KNY secret-sharing scheme uses a witness encryption scheme for NP and a non-interactive commitment scheme with perfect binding. The latter primitive can be constructed from one-way permutations [98].[27]

[26] As we define security with respect to non-uniform polynomial-time adversaries, which are deterministic, it is not too hard to formalized the arguments below. In [116] they consider (uniform) randomized polynomial-time adversaries and formalizing the proof requires more details.

[27] It can also be constructed in the CRS model from one-way functions [139]. We only state the results for the plain model.

Theorem 9.12 ([116])**.** *If witness encryption schemes for* NP *exist and one-way permutations exist, then there is a computational secret-sharing scheme for circuits computing monotone functions.*

9.5 A Provable Separation Between Information-Theoretic and Computational Secret-Sharing Schemes

We next present a construction from [7] showing that computational secret-sharing schemes are provably more efficient than information-theoretic secret-sharing schemes even for a one-bit secret – we describe a computational secret-sharing scheme realizing the Csirmaz access structure Γ^n_{Csi} (defined in Definition 7.7) with max share size $O(\lambda)$; in contrast, every information-theoretic secret-sharing scheme realizing Γ^n_{Csi} requires shares of size $\Omega(n/\log n)$ (see Theorem 7.8). The construction uses the minimal assumption that one-way functions exist.

The computational secret-sharing scheme realizing $\left\langle \Gamma^n_{\text{Csi}} \right\rangle_{n\in\mathbb{N}}$ is described in Figure 9.7. We start with an informal description of the scheme. Recall that the minimal authorized sets of Γ^n_{Csi} are $\{p_i\} \cup A_i$, where $A_1, \ldots, A_{2^k}$ are subsets of a set A of size $k = O(\log(n))$ (in some fixed order) and $1 \leq i \leq 2^k$. In an information-theoretic secret-sharing scheme realizing Γ^n_{Csi} we can share the secret $s \in \{0,1\}$ independently for every minimal authorized set, that is, for every authorized set $\{p_i\} \cup A_i$ the dealer chooses random bits $\left\langle r_{i,j} \right\rangle_{p_j \in A_i}$ and gives $r_{i,j}$ to p_j and gives $s \oplus \bigoplus_{p_j \in A} r_{i,j}$ to p_i. In other words, the share of each $p_j \in A$ is the random bits $\left\langle r_{i,j} \right\rangle_{p_j \in A_i}$ and the share of p_j for $1 \leq j \leq 2^k$ is a bit determined by the bits given to the parties in A and the secret. In the computational secret-sharing scheme realizing Γ^n_{Csi}, the bits given to $p_j \in A$ will be pseudorandom bits generated by a seed of a pseudorandom generator PRG given to p_j, the bits of the parties not in A will be computed from these pseudorandom bits.

A computational secret-sharing scheme realizing $\left\langle \Gamma^n_{\text{Csi}} \right\rangle_{n\in\mathbb{N}}$

The secret: a string $s \in \{0,1\}$.
The scheme:

- For every $p_j \in A$ choose with uniform distribution a seed $a_j \in \{0,1\}^\lambda$. Let $\left\langle r_{1,j}, \ldots, r_{2^k,j} \right\rangle \leftarrow \mathsf{PRG}(a_j)$.
- **Share of party $p_j \in A$.** The seed a_j.
- **Share of party p_i for $1 \leq i \leq 2^k$.** The bit $s \oplus \bigoplus_{p_j \in A_i} r_{i,j}$.

Figure 9.7: A computational secret-sharing scheme realizing $\left\langle \Gamma^n_{\text{Csi}} \right\rangle_{n\in\mathbb{N}}$ using a pseudorandom generator PRG that stretches seeds of length λ to pseudorandom strings of length $2^k \leq n$.

The efficient reconstruction of the secret by a minimal authorized set $\{p_i\} \cup A_i$ is obvious: for each

$p_j \in A_i$ compute $r_{i,j}$ by applying PRG to a_j and reconstruct s from $s \oplus \bigoplus_{p_j \in A_i} r_{i,j}$ – the share of p_i. The computational security of the scheme follows from a simple hybrid argument, see [7]. As one-way functions imply pseudorandom generators [104], we obtain the following theorem.

Theorem 9.13 ([7])**.** *If a one-way functions exist, then there is a computational secret-sharing scheme realizing* $\left\langle \Gamma^n_{\text{Csi}} \right\rangle_{n\in\mathbb{N}}$ *in which the share size is* λ*, the security parameter.*

Remark 9.14. In [7], it is shown that there is an access structure that requires *total* share size $\Omega(n^2/\log(n))$ and can be realized by a computational secret-sharing scheme with share size $O(\lambda)$ – the security parameter; this access structure is a variant of Γ^n_{CsiTot} (described in Definition 7.10). This separation is nearly the best possible separation between information-theoretic a computational secret-sharing schemes with the currently known lower bounds.

9.5.1 Succinct Computational Secret-Sharing Schemes

In the Yao's and KNY schemes, the share size is polynomial in the representation of the access structure (by monotone circuits and general circuits, respectively). Applebaum et al. [7] raised the question wether there are computational secret-sharing schemes in which the share size is much smaller than the representation, e.g., logarithmic in the representation; such schemes are called succinct. Applebaum et al. showed that, under the RSA assumption, there are computational secret-sharing schemes for access structures represented by CNF formulas, where the share size is $\text{poly}(n, \log(\ell))$, where n is the number of parties in the access structure and ℓ is the number of clauses in the CNF formula. This result implies that, under the sub-exponential RSA assumption, every access structure can be realized by a computational secret-sharing scheme with share size polynomial in the number of parties (using a CNF formula with at most 2^n clauses to represent the access structure). The running time of the sharing and reconstruction in the scheme of [7] for general access structures is exponential; however, results of [121] imply that this is unavoidable. In this monograph, we will only describe the high-level idea of the construction of [7].

The main tool for constructing these schemes is a new cryptographic primitive called a projective pseudorandom generator (abbreviated pPRG). A pPRG expands a short seed into a longer pseudorandom string for which any subset of the bits of the pseudorandom string can be revealed without disclosing any information about the other bits of the string. Of course, this can be accomplished by simply giving the subset of the output bits; however, we require that this is done using a short projective key (or seed).

Applebaum et al. construct a pPRG based on the RSA assumption in which the size of the projective key is polylogarithmic in the number of bits in the output of the pPRG. We will not describe the construction in this monograph. Given a pPRG, Applebaum et al. construct a secret sharing scheme for an access structure represented by a CNF formula $\varphi(y_1, \ldots, y_n)$. The idea for this construction is to start with the information-theoretic secret-sharing scheme of [107] for CNF formulas, described in Figure 4.3, and use pseudorandom bits in it instead of random bits. In particular, in the construction of Figure 4.3 for a formula $\varphi(y_1, \ldots, y_n)$ with ℓ clauses, there is a random bit for r_i each of the first $\ell - 1$ clauses of the formula and a bit r_ℓ for the last clause, which is the exclusive-OR of the $\ell - 1$ random bits and the secret. The share sh_i contains all

bits of clauses that contain y_i. Given a satisfying assignment $x \in \{0,1\}^n$, in each clause there is at least one variable that is satisfied by x, meaning that the shares $\langle \mathsf{sh}_i \rangle_{i:x_i=1}$ contain the ℓ bits of the clauses, and therefore the secret can be recovered from them. Note that in the above scheme, there are $\ell - 1$ random bits and each share contains a subset of them (and possibly also contains r_ℓ). This is exactly the functionality provided by a pPRG — the $\ell - 1$ bits will be the output of the pPRG and the share of a party is the projective key for the appropriate set; if y_i appears in the last clause then, in addition, the share of p_i also contains r_ℓ. In particular, the size of the shares is determined by the length of the projective keys.

Chapter 10

Summary and Open Problems

In this monograph, we considered secret-sharing schemes, a basic tool in cryptography that has many applications. We mostly considered information-theoretic secret-sharing schemes, i.e., schemes that are secure even against an unbounded adversary that tries to break them. These are fairly simple schemes (e.g., unlike MPC protocols, there is no interaction). Nevertheless, their complexity (i.e., share size) is not understood. Studying secret-sharing schemes can be a first step in understanding more complex information-theoretic cryptographic primitives. In the rest of the chapter, we summarize the material covered in this monograph, mention two subjects not covered in this monograph, and describe some open problems.

10.1 Summary of the Subjects Covered in This Monograph

We started this monograph by discussing the most useful secret-sharing schemes – threshold t-out-of-n secret-sharing schemes; these schemes are widely used in cryptography, i.e., for constructing secure multiparty computation (MPC) protocols for arbitrary functionalities [97, 34] (see Chapter 6). We showed Shamir's t-out-of-n secret-sharing scheme [163], which is based on polynomials (i.e., on Reed-Solomon error correcting codes). Shamir's scheme is ideal when the size of the secrets is greater than $\log(n)$, i.e., the shares and the secret have the same size. We proved that having shares of size $\log(n)$ is unavoidable as even sharing 1-bit secrets requires shares of size at least $\log(n)$ [114, 47]. We then discussed ramp secret-sharing schemes [42], in which sets of size t can reconstruct the secret, while sets of size at most b cannot learn any information on the secret, for some $1 \leq b < t \leq n$. We have shown that in such schemes the share can be as small as $1/(t-b)$ times the size of the secret [42]. Furthermore, when $t - b = \theta(n)$, the share size can be $O(1)$ [58, 59].

We then defined secret-sharing schemes for general access structures, giving two equivalent definitions, one that does not assume a distribution on the secret and the second that assumes such distribution. We argued that for cryptographic applications, the first definition is more suitable; however, for proving lower bounds, the second definition is preferable. We showed several constructions of secret-sharing schemes, starting from the scheme of [107], based on DNF and CNF formulas. We then described its generalization

by [36], showing that if an access structure can be represented by a small monotone formula, then it has an efficient secret-sharing scheme. We continued by showing the construction of secret-sharing schemes from monotone span programs [49, 111]. Monotone span programs are equivalent to linear secret-sharing schemes and are equivalent to schemes where the reconstruction is linear [14]. As every monotone formula can be transformed into a monotone span program of the same size, the monotone span program construction is a generalization of the construction of [107, 36]. Furthermore, there are functions that have small monotone span programs and do not have small monotone formulas [11]; thus, this is a strict generalization. Finally, we presented the multilinear construction of secret-sharing schemes. We remark that the linearity of a scheme is important in many applications, as we demonstrated in Chapter 6 for the construction of secure multiparty protocols for general functions.

In all the secret-sharing schemes constructed until 2018, the share size for almost all n-party access structures was $2^{(1-o(1))n}$. In an impressive result, Liu and Vaikuntanathan [123] (using results of [124, 125]) constructed for every n-party access structure, secret-sharing schemes with share size $2^{0.994n}$. This was improved in a sequence of works [6, 8, 9], where the currently best known scheme has share size $(3/2)^{(1+o(1))n} < 2^{0.585n}$ [9]. In this monograph, we gave ideas of these constructions and described a secret-sharing scheme for arbitrary access structure with share size 2^{cn} for some $0.9 < c < 1$. The construction is self-contained (except for two simple probabilistic claims).

Even in the recent secret-sharing schemes, the share size for the worst n-party access structure is exponential in n. The best known lower bounds on the share size are from the nineties and are far from the best known upper bounds – Csirmaz [66, 67] showed that for every n there is an n-party access structure that requires total information ratio $\Omega(n^2/\log(n))$ in any secret-sharing scheme realizing it, i.e., for every ℓ in any secret-sharing scheme realizing it with secrets of length ℓ the share size is $\Omega((n^2/\log(n)) \cdot \ell)$. In this monograph, we provided a proof of this lower bound. We then discussed lower bounds for linear secret-sharing schemes – in this case it is known that for every n there are explicit n-party access structures that in any linear secret-sharing scheme realizing them the information ratio is $2^{\Omega(n)}$ [150]. In this monograph, we provided a proof of a weaker lower bound, namely $n^{\Omega(\log(n))}$. We also proved, based on [11], that for almost all access structures the share size in every linear secret-sharing scheme realizing them is at least $2^{(0.5-o(1))n}$.

In any secret-sharing scheme, the size of each share of any non-redundant party is at least the size of the secret [112]. An access structure is ideal if it can be realized by secret-sharing schemes in which the share size of each party is the size of the secret, that is, by a scheme in which the share size is the minimal possible. The characterization of the ideal access structures is partially given via matroids, combinatorial structures that abstract and generalize the notion of linear independence in vector spaces and spanning trees in graphs. Matroids were defined in 1935 [184], long before the introduction of secret-sharing schemes; it is quite interesting that these two objects are related. In this monograph, we proved results of [50] showing that (1) if an access structure is ideal then the access structure is a port of a matroid, and (2) if an access structure is a port of a *linear or multilinear* matroid, then the access structure is ideal. These two results give a partial characterization of ideal access structures; their exact characterization is not known. In particular, it is not known if there is an ideal access structure that does not have an ideal multilinear secret-sharing scheme.

In all the results we mentioned so far in this chapter, the security was information-theoretic, i.e., an unbounded adversary cannot learn any information on the secret. To decrease the share size we also considered computational secret-sharing schemes in which the security only holds against a polynomial-time adversary (as common in cryptography). Furthermore, in computational secret-sharing schemes we also require that the sharing and reconstruction algorithms run in polynomial time. We described four constructions of computational secret-sharing schemes, showing that they can be more efficient than information-theoretic secret-sharing schemes: (1) a threshold t-out-of-n secret-sharing scheme of [117] with information ratio $O(1/t)$, (2) a secret-sharing scheme of Yao [186] for every access structure whose share size is the size of a *monotone circuit* representing the access structure, (3) a secret-sharing scheme of Komargodski et al. [116] for every monotone access structure in which the share size is polynomial in the size of a *non-monotone* circuit representing the access structure, and (4) a secret-sharing scheme of [7] for the Csirmaz access structure in which the share size is $O(\lambda)$, where λ is the security parameter.

A recent result of Applebaum et al. [7] showed that every access structure can be realized by a computational secret-sharing scheme with polynomial share size (under the RSA assumption). The running time of the sharing and reconstruction in the scheme of [7] is exponential; however, results of [121] imply that this is unavoidable. This result is not described in this monograph.

10.2 Some Subjects Not Covered in This Monograph

Obviously, we could not cover all the results on secret-sharing schemes in this monograph. Some results are too advanced (we tried to at least mention them in the monograph), some areas are not mature enough to be covered here, and some do not fit in the flow of the results we described. We next mention two omissions.

Verifiable and Robust Secret-Sharing Schemes. In all the schemes we presented in this monograph, we assumed that all participants – the dealer and the parties – are honest. Secret-sharing schemes without these assumptions were studied and used to construct secure multiparty computation (MPC) protocols that are secure against malicious parties (which can send arbitrary messages).

The simpler scenario is when the dealer is honest; however, in the reconstruction of the secret by some authorized set, some parties might submit incorrect shares. Robust secret-sharing schemes, studied e.g., in [175, 153] and many follow-up works, provide security against such cheating parties. They can be constructed from regular secret-sharing schemes by adding message authentication to the shares and giving authentication keys to the parties.

In the more complex scenario, the dealer is corrupt while sharing the secret and some parties are corrupt while reconstructing the secret. A verifiable secret-sharing scheme (VSS), defined by Chor et al. [61] and studied in many follow-up papers, handles these two issues. VSS is used in secure multiparty computation (MPC) protocols secure against malicious parties (e.g., [153]). More details on VSS can be found in the monographs of Chandramouli, Choudhury, and Patra [55] and Krenn and Lorünser [118].

Leakage Resilient Secret Sharing. Using side-channel attacks, the adversary might get shares of an unauthorized set and some bounded information on the shares of other parties. A leakage-resilient secret-sharing scheme, introduced by Goyal and Kumar [100] and Benhamouda, Degwekar, Ishai, and Rabin [37], is a secret-sharing scheme that is secure against such an adversary; that is, an adversary that obtains shares of any unauthorized subset of parties along with bounded leakage from the other shares learns no information about the secret. Leakage-resilient secret-sharing schemes have been thoroughly studied in recent years, both in the information-theoretic setting and in the computational setting (see, e.g., [115] for a list of such works). The works study various leakage models, such as static adversaries vs. adaptive adversaries (where leakage queries depend on prior leakage responses), local leakage vs. joint (combined leakage from multiple shares) leakages, various leakage functions, and threshold secret-sharing schemes vs. secret-sharing schemes for arbitrary access structures.

10.3 Open Problems

10.3.1 Secret-Sharing Schemes for Arbitrary Access Structures

An important open problem regarding secret-sharing schemes is settling the optimal share size of secret-sharing schemes for arbitrary access structures for a one-bit secret. That is,

Question 10.1. *What is the minimal share size for sharing a one-bit secret in a scheme realizing the worst n-party access structure? Is it $2^{\Omega(n)}$? Is it* $\mathrm{poly}(n)$*?*

Consistent with our current knowledge, the share size can be anywhere between the above two bounds. We do not even have some unexpected consequences of the share size being polynomial. In the preliminary version of this monograph [15], I conjectured that the share size must be exponential. Due to the new constructions of secret-sharing schemes and conditional disclosure of secrets protocols [124, 125, 123, 6, 8, 19, 9], I do not make any conjectures on the share size.

Even if sharing a one bit secret requires exponential share size, it is possible that the information ratio of every access structure is small (for very long secrets). For example, Applebaum and Arkis [4] proved that $2^{2^{n/2}}$ access structures with n parties can be realized with information rate 4; the length of the secrets in their scheme is large, i.e., $2^{2^{n/2}}$. Such schemes with constant information ratio are not known for short secrets. It is interesting to determine if secret-sharing schemes with small information ratio exist for all access structures.

Question 10.2. *What is the minimal information ratio in a scheme realizing the worst n-party access structure? Is it $2^{\Omega(n)}$? Is it* $\mathrm{poly}(n)$*?*

The above two questions are open even for non-explicit access structures.

10.3.2 Linear Secret-Sharing Schemes for Arbitrary Access Structures

For linear secret-sharing schemes, there is also a gap between the best known upper bound on the share size and the best known lower bound; however, this gap is much smaller than the gap for general secret-sharing

schemes. The best known upper bound for linear secret-sharing schemes for arbitrary access structures is $2^{0.7563n}$ [2], while the best known lower bound for explicit access structures is 2^{cn} for some constant $0 < c < 1$ [150] and the best known lower bound for almost all n-party access is $2^{0.5n-o(n)}$ [11]. The question of whether this bound is tight is open.

Question 10.3. *Determine the smallest constant c such that every n-party access structure can be realized by a linear secret-sharing scheme with share size $2^{cn+o(n)}$. In particular, can every n-party access structure be realized by a linear secret-sharing scheme with share size $2^{0.5n+o(n)}$?*

In this monograph, we described linear and multilinear secret-sharing schemes. It is known that multilinear schemes are exponentially more efficient than linear schemes for many access structures [4]. On the other hand, there is an explicit access structure such that every multilinear secret-sharing scheme that realizes it has information rate $n^{\Omega(\log(n))}$ [17]. Proving exponential lower bounds for multilinear secret-sharing schemes or constructing multilinear secret-sharing schemes with sub-exponential share size for arbitrary access structures is open. As we do not know the optimal share size of general general secret-sharing schemes for arbitrary access structures, proving "strong" lower bounds for a large class of secret-sharing schemes is desirable.

Question 10.4. *What is the minimal information ratio of multilinear secret-sharing schemes realizing the worst n-party access structure? Is it $2^{\Omega(n)}$? Is it $n^{\Omega(\log(n))}$?*

10.3.3 Efficient Secret-Sharing Schemes

Possibly the question that would have the most practical applications is constructing new efficient secret-sharing schemes, that is, secret-sharing schemes in which the share size is polynomial in the number of parties. Most constructions of efficient secret-sharing schemes are linear; the access structures that can be realized by efficient linear secret-sharing schemes are the access structures that have polynomial-size monotone span programs [111]. There are a few examples of access structures that have efficient non-linear secret-sharing schemes and do not have efficient linear secret-sharing schemes, e.g., the constructions of [24, 178, 31]. There are new non-linear constructions of secret-sharing schemes for arbitrary access structures [124, 125, 123, 6, 8, 9]; however, their share size is not polynomial. They do imply secret-sharing schemes with polynomial share size for k-slice access structures for $k \leq \log(n)/(\log\log(n))^2$ [6] and $k > n - \log(n)/(\log\log(n))^2$ [20].[28] It is interesting to construct new non-linear secret-sharing schemes.

Question 10.5. *Construct efficient non-linear secret-sharing schemes for a larger family of access structures than the access structures that have efficient linear secret-sharing schemes (for short secrets).*

In particular, are there efficient information-theoretic secret-sharing schemes for access structures represented by polynomial-size monotone circuits, or even by polynomial-size non-monotone circuits (the access

[28] A k-slice access structure is an access structure in which all sets of size at least $k + 1$ are authorized, all sets of size at most $k - 1$ are unauthorized, and each set of size k can be either authorized or unauthorized.

structure is of course monotone)? Recall that we described efficient computational secret-sharing schemes for these access structures (under some hardness assumptions).

10.3.4 Secret-Sharing Schemes for Natural Access Structures

There are interesting access structures that we do not know whether they have efficient secret-sharing schemes with information-theoretic security. The first access structure is the *directed connectivity* access structure whose parties are edges in a complete directed graph and whose authorized sets are sets of edges containing a path from v_1 to v_m. As there is a small monotone circuit for this access structure, by [186] (see Theorem 9.6) it has an efficient computational scheme. In [28], it was proved that in every *linear* secret-sharing scheme realizing the directed connectivity access structure the size of the shares is $n^{\Omega(\log(n))}$. It is not known if the directed connectivity access structure has an efficient (non-linear) secret-sharing scheme. In comparison, the *undirected connectivity* access structure has an efficient perfect scheme [35] (see Section 4.1).

The second access structure that we do not know if it has an efficient scheme is the *perfect matching* access structure, described in Definition 9.9. The parties of this access structure are edges in a complete undirected graph and the authorized sets are sets of edges containing a perfect matching. By [116] (see Theorem 9.12), this access structure has a computational secret-sharing scheme assuming the existence of witness encryption schemes for NP. It is open if this access structure has an efficient information-theoretic secret-sharing scheme or even if it has an efficient computational scheme assuming the existence of one-way functions (as every monotone circuit for perfect matching has super-polynomial size [156], the construction of [186] will not work). We remark that an efficient scheme for this access structure implies an efficient (computational or information-theoretic) scheme for the directed connectivity access structure. In particular, by [28], in every *linear* secret-sharing scheme realizing the perfect matching access structure the size of the shares is $n^{\Omega(\log(n))}$.

The third interesting family of access structures is *weighted threshold* access structures, already considered by Shamir [163]. In such an access structure, each party has a weight and there is some threshold. A set of parties is authorized if and only if the sum of the weights of the parties in the set is bigger than the threshold. In recent years, weighted threshold access structures have gained attention [38, 92, 74, 176] as they are motivated by stake-based blockchains [113, 48], where different users have different stakes. For these access structures there is an efficient computational scheme [30] and a perfect information-theoretic secret-sharing scheme with shares of size $n^{O(\log(n))}$ [30]. In [38, 92], more efficient weighted threshold secret-sharing schemes were constructed assuming there is a gap between the weight of authorized sets and the weight of unauthorized sets. In [81], secret-sharing schemes with a different notion of approximation of weighted access structures were presented. It is open if weighted threshold access structures (without a gap) have a perfect secret-sharing scheme with polynomial-size shares. Furthermore, it is open if they can be represented by polynomial-size monotone formulas.

Finally, the fourth interesting family of access structures is graph access structures, introduced in [50]; we discussed graph access structures in Chapter 5, using them as a building block to construct secret-sharing schemes for arbitrary access structures. In a graph secret-sharing scheme, the parties are vertices of a graph

and a set of vertices (parties) can reconstruct the secret if and only if it contains an edge. In other words, all minimal authorized sets are of size 2 and a set is unauthorized if it is an independent set in the graph. Graph secret-sharing schemes were studied in many papers, e.g., [50, 51, 53, 77, 43, 68, 73, 69, 70, 21, 82, 72]. The naive scheme to realize a graph is to share the secret independently for each edge; this result implies a share of size $O(n)$ per party for an n-vertex graph. A better scheme with share size $O(n/\log(n))$ per party is implied by a result of Erdös and Pyber [80]. In contrast, the best lower bounds on the share size of secret-sharing schemes for graphs is $\Omega(\log(n))$ and the best known lower bound on the total share size is $\Omega(n\log(n))$ [77, 68]; this lower bound holds for the d-dimensional hypercube with $n = 2^d$ vertices. Despite the improvements in the share size for general access structures and for slice access structures, there was no major improvement in the share size of secret-sharing schemes for graphs. Understanding the share size required to realize graph access structures is a step towards understanding the share size required for general access structures.

Question 10.6. *What is the share size required to realize an arbitrary n-vertex graph access structure?*

Acknowledgment

I would like to thank my co-authors in papers related to secret-sharing: Damiano Abram, Bar Alon, Benny Applebaum, Tamar Ben David, Aner Ben-Efraim, Mike Burmester, Benny Chor, Yvo Desmedt, Eran Omri, Oriol Farràs, Matt Franklin, Anna Gál, Yuval Ishai, Eyal Kushilevitz, Or Lasri, Noam Livne, Tianren Liu, Varun Narayanan, Oded Nir, Yuval Mintz, Ilan Orlov, Hussien Othman, Carles Padró, Anat Paskin-Cherniavsky, Nati Peter, Mike Paterson, Toniann Pitassi, Tamir Tassa, Ilya Tyomkin, Vinod Vaikuntanathan, and Enav Weinreb. I learned a lot from working with them. I would like to thank Idan Saltzman and Ofek Yabo for reading parts of this monograph and providing suggestions for improving it.

Bibliography

[1] Damiano Abram, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Varun Narayanan. Cryptography from planted graphs: Security with logarithmic-size messages. In Guy N. Rothblum and Hoeteck Wee, editors, *Twentyth Theory of Cryptography Conference – TCC 2023*, volume 14369 of *LNCS*, pages 286–315. Springer, 2023.

[2] Bar Alon, Amos Beimel, and Or Lasri. Simplified PIR and CDS protocols and improved linear secret-sharing schemes. Technical Report 2024/1599, IACR Cryptol. ePrint Arch., 2024.

[3] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3:289–304, 1992.

[4] Benny Applebaum and Barak Arkis. On the power of amortization in secret sharing: d-uniform secret sharing and CDS with constant information rate. *ACM Trans. Comput. Theory*, 12(4):24:1–24:21, 2020.

[5] Benny Applebaum, Barak Arkis, Pavel Raykov, and Prashant Nalini Vasudevan. Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. *SIAM J. Comput.*, 50(1):32–67, 2021.

[6] Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 441–471, 2019.

[7] Benny Applebaum, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, and Vinod Vaikuntanathan. Succinct computational secret sharing. In *55th STOC*, pages 1553–1566, 2023.

[8] Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. In *52nd STOC*, pages 280–293, 2020.

[9] Benny Applebaum and Oded Nir. Upslices, downslices, and secret-sharing with complexity of 1.5^n. In *CRYPTO 2021*, volume 12827 of *LNCS*, pages 627–655, 2021.

[10] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577, 2014.

[11] László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

[12] Michael Bamiloshin, Aner Ben-Efraim, Oriol Farràs, and Carles Padró. Common information, matroid representation, and secret sharing for matroid ports. *Des. Codes Cryptogr.*, 89(1):143–166, 2021.

[13] Ohad Barta, Yuval Ishai, Rafail Ostrovsky, and David J. Wu. On succinct arguments and witness encryption from groups. In *CRYPTO 2020*, volume 12170 of *LNCS*, pages 776–806, 2020.

[14] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion, 1996. `www.cs.bgu.ac.il/~beimel/pub.html`.

[15] Amos Beimel. Secret-sharing schemes: A survey. In *IWCC 2011*, volume 6639 of *LNCS*, pages 11–46, 2011.

[16] Amos Beimel. Lower bounds for secret-sharing schemes for k-hypergraphs. In *ITC 2023*, volume 267 of *LIPIcs*, pages 16:1–16:13, 2023.

[17] Amos Beimel, Aner Ben-Efraim, Carles Padró, and Ilya Tyomkin. Multi-linear secret-sharing schemes. In *TCC 2014*, volume 8349 of *LNCS*, pages 394–418, 2014.

[18] Amos Beimel and Benny Chor. Universally ideal secret-sharing schemes. *IEEE Trans. on Information Theory*, 40(3):786–794, 1994.

[19] Amos Beimel and Oriol Farràs. The share size of secret-sharing schemes for almost all access structures and graphs. In *TCC 2020*, volume 12552 of *LNCS*, pages 499–529, 2020.

[20] Amos Beimel, Oriol Farràs, Or Lasri, and Oded Nir. Secret-sharing schemes for high slices. In Elette Boyle and Mohammad Mahmoody, editors, *21th Theory of Cryptography Conference – TCC 2024*, volume 15367 of *LNCS*, pages 581–613. Springer, 2024.

[21] Amos Beimel, Oriol Farràs, and Yuval Mintz. Secret-sharing schemes for very dense graphs. *J. of Cryptology*, 29(2):336–362, 2016.

[22] Amos Beimel, Oriol Farràs, Yuval Mintz, and Naty Peter. Linear secret-sharing schemes for forbidden graph access structures. *IEEE Trans. Inf. Theory*, 68(3):2083–2100, 2022.

[23] Amos Beimel, Anna Gál, and Mike Paterson. Lower bounds for monotone span programs. *Computational Complexity*, 6(1):29–45, 1997. Conference version: FOCS '95.

[24] Amos Beimel and Yuval Ishai. On the power of nonlinear secret-sharing. *SIAM J. on Discrete Mathematics*, 19(1):258–280, 2005.

[25] Amos Beimel and Noam Livne. On matroids and nonideal secret sharing. *IEEE Trans. Inf. Theory*, 54(6):2626–2643, 2008.

[26] Amos Beimel, Noam Livne, and Carles Padró. Matroids can be far from ideal secret sharing. In *TCC 2008*, volume 4948 of *LNCS*, pages 194–212, 2008.

[27] Amos Beimel and Ilan Orlov. Secret sharing and non-shannon information inequalities. *IEEE Trans. on Information Theory*, 57(9):5634–5649, 2011.

[28] Amos Beimel and Anat Paskin. On linear secret sharing for connectivity in directed graphs. In *Sixth SCN*, volume 5229 of *LNCS*, pages 172–184, 2008.

[29] Amos Beimel, Tamir Tassa, and Enav Weinreb. Characterizing ideal weighted threshold secret sharing. *SIAM J. Discret. Math.*, 22(1):360–397, 2008.

[30] Amos Beimel and Enav Weinreb. Monotone circuits for weighted threshold functions. In *20th CCC*, 2005.

[31] Amos Beimel and Enav Weinreb. Separating the power of monotone span programs over different fields. *SIAM J. on Computing*, 34(5):1196–1215, 2005.

[32] Mihir Bellare and Phillip Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. Technical Report 2006/449, Cryptology ePrint Archive, 2006. `eprint.iacr.org/`.

[33] Aner Ben-Efraim. Secret-sharing matroids need not be algebraic. *Discret. Math.*, 339(8):2136–2145, 2016.

[34] Michael Ben-Or, Shaffi Goldwasser, and Avi Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *20th STOC*, pages 1–10, 1988.

[35] Josh Benaloh and Steven Rudich. Private communication, 1989.

[36] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *CRYPTO '88*, volume 403 of *LNCS*, pages 27–35, 1988.

[37] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018*, volume 10991 of *LNCS*, pages 531–561. Springer, 2018.

[38] Fabrice Benhamouda, Shai Halevi, and Lev Stambler. Weighted secret sharing from wiretap channels. In *ITC 2023*, volume 267 of *LIPIcs*, pages 8:1–8:19, 2023.

[39] Michael Bertilsson and Ingemar Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In *AUSCRYPT '92*, volume 718 of *LNCS*, pages 67–79, 1992.

[40] George Robert Blakley. Safeguarding cryptographic keys. In *1979 AFIPS National Computer Conference*, volume 48, pages 313–317, 1979.

[41] George Robert Blakley and Grigory A. Kabatianskii. Linear algebra approach to secret sharing schemes. In *Error Control, Cryptology, and Speech Compression*, volume 829 of *LNCS*, pages 33–40, 1994.

[42] George Robert Blakley and Catherine A. Meadows. Security of ramp schemes. In *CRYPTO '84*, volume 196 of *LNCS*, pages 242–268, 1984.

[43] Carlo Blundo, Alfredo De Santis, Roberto De Simone, and Ugo Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Designs, Codes and Cryptography*, 11(2):107–122, 1997.

[44] Carlo Blundo, Alfredo De Santis, Luisa Gargano, and Ugo Vaccaro. On the information rate of secret sharing schemes. *Theoretical Computer Science*, 154(2):283–306, 1996.

[45] Carlo Blundo, Alfredo De Santis, and Ugo Vaccaro. On secret sharing schemes. *Inform. Process. Lett.*, 65(1):25–32, 1998.

[46] Andrej Bogdanov. Csirmaz's duality conjecture and threshold secret sharing. In *ITC 2023*, volume 267 of *LIPIcs*, pages 3:1–3:6, 2023.

[47] Andrej Bogdanov, Siyao Guo, and Ilan Komargodski. Threshold secret sharing requires a linear size alphabet. In *Fourteenth Theory of Cryptography Conference – TCC 2016-B*, volume 9986 of *LNCS*, pages 471–484, 2016.

[48] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, and Fan Zhang. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. Technical report, Chainlink Labs, 2021.

[49] Ernest F. Brickell. Some ideal secret sharing schemes. *Journal of Combin. Math. and Combin. Comput.*, 6:105–113, 1989.

[50] Ernest F. Brickell and Daniel M. Davenport. On the classification of ideal secret sharing schemes. *J. of Cryptology*, 4(73):123–134, 1991.

[51] Ernest F. Brickell and Douglas R. Stinson. Some improved bounds on the information rate of perfect secret sharing schemes. *J. of Cryptology*, 5(3):153–166, 1992.

[52] Christian Cachin. On-line secret sharing. In *Cryptography and Coding, 5th IMA Conference*, volume 1025 of *LNCS*, pages 190–198. Springer, 1995.

[53] Renato M. Capocelli, Alfredo De Santis, Luisa Gargano, and Ugo Vaccaro. On the size of shares for secret sharing schemes. *J. of Cryptology*, 6(3):157–168, 1993.

[54] Ignacio Cascudo Pueyo, Ronald Cramer, and Chaoping Xing. Bounds on the threshold gap in secret sharing and its applications. *IEEE Trans. on Information Theory*, 59(9):5600–5612, 2013.

[55] Anirudh Chandramouli, Ashish Choudhury, and Arpita Patra. A survey on perfectly secure verifiable secret-sharing. *ACM Comput. Surv.*, 54(11s):232:1–232:36, 2022.

[56] Arup Kumar Chattopadhyay, Sanchita Saha, Amitava Nag, and Sukumar Nandi. Secret sharing: A comprehensive survey, taxonomy and applications. *Computer Science Review*, 51:100608, 2024.

[57] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In *20th STOC*, pages 11–19, 1988.

[58] Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 521–536. Springer, 2006.

[59] Hao Chen, Ronald Cramer, Shafi Goldwasser, Robbert de Haan, and Vinod Vaikuntanathan. Secure computation from random error correcting codes. In *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 291–310, 2007.

[60] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In *CRYPTO 2018*, volume 10992 of *LNCS*, pages 577–607, 2018.

[61] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *26th FOCS*, pages 383–395, 1985.

[62] Benny Chor and Eyal Kushilevitz. Secret sharing over infinite domains. *J. of Cryptology*, 6(2):87–96, 1993.

[63] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[64] Ronald Cramer, Ivan Damgård, and Ueli Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 316–334, 2000.

[65] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[66] László Csirmaz. The size of a share must be large. In *EUROCRYPT '94*, volume 950 of *LNCS*, pages 13–22, 1994.

[67] László Csirmaz. The dealer's random bits in perfect secret sharing schemes. *Studia Sci. Math. Hungar.*, 32(3–4):429–437, 1996.

[68] László Csirmaz. Secret sharing schemes on graphs. Technical Report 2005/059, Cryptology ePrint Archive, 2005. `eprint.iacr.org/`.

[69] László Csirmaz. An impossibility result on graph secret sharing. *Des. Codes Cryptography*, 53(3):195–209, 2009.

[70] László Csirmaz. Secret sharing on the d-dimensional cube. *Designs, Codes and Cryptography*, 74(3):719–729, 2015.

[71] László Csirmaz. Secret sharing and duality. *J. Math. Cryptol.*, 15(1):157–173, 2021.

[72] László Csirmaz and Péter Ligeti. Secret sharing on large girth graphs. *Cryptogr. Commun.*, 11(3):399–410, 2019.

[73] László Csirmaz and Gábor Tardos. Optimal information rate of secret sharing schemes on trees. *IEEE Trans. Inf. Theory*, 59(4):2527–2530, 2013.

[74] Sourav Das, Benny Pinkas, Alin Tomescu, and Zhuolun Xiang. Distributed randomness using weighted VUFs. Cryptology ePrint Archive, Paper 2024/198, 2024.

[75] Yvo Desmedt and Yair Frankel. Shared generation of authenticators and signatures. In *CRYPTO '91*, volume 576 of *LNCS*, pages 457–469, 1991.

[76] Marten van Dijk. A linear construction of perfect secret sharing schemes. In *EUROCRYPT '94*, volume 950 of *LNCS*, pages 23–34, 1995.

[77] Marten van Dijk. On the information rate of perfect secret sharing schemes. *Des. Codes Cryptography*, 6(2):143–169, 1995.

[78] Randall Dougherty, Christopher F. Freiling, and Kenneth Zeger. Six new non-Shannon information inequalities. In *ISIT 2006*, pages 233–236, 2006.

[79] Randall Dougherty, Christopher F. Freiling, and Kenneth Zeger. Non-shannon information inequalities in four random variables. *CoRR*, abs/1104.3602, 2011.

[80] Paul Erdös and László Pyber. Covering a graph by complete bipartite graphs. *Discrete Mathematics*, 170(1–3):249–251, 1997.

[81] Oriol Farràs and Miquel Guiot. Reducing the share size of weighted threshold secret sharing schemes via chow parameters approximation. In Elette Boyle and Mohammad Mahmoody, editors, *21th Theory of Cryptography Conference – TCC 2024*, volume 15367 of *LNCS*, pages 517–547. Springer, 2024.

[82] Oriol Farràs, Tarik Kaced, Sebastià Martín, and Carles Padró. Improving the linear programming technique in the search for lower bounds in secret sharing. *IEEE Trans. Inf. Theory*, 66(11):7088–7100, 2020.

[83] Oriol Farràs, Jaume Martí-Farré, and Carles Padró. Ideal multipartite secret sharing schemes. *J. Cryptol.*, 25(3):434–463, 2012.

[84] Oriol Farràs, Jessica Ruth Metcalf-Burton, Carles Padró, and Leonor Vázquez. On the optimization of bipartite secret sharing schemes. *Des. Codes Cryptogr.*, 63(2):255–271, 2012.

[85] Oriol Farràs and Carles Padró. Ideal hierarchical secret sharing schemes. *IEEE Transactions on Information Theory*, 58(5):3273–3286, 2012.

[86] Serge Fehr. Efficient construction of the dual span program. Manuscript, 1999.

[87] Matthew Franklin and Moti Yung. Communication complexity of secure computation. In *24th STOC*, pages 699–710, 1992.

[88] Satoru Fujishige. Polymatroidal dependence structure of a set of random variables. *Information and Control*, 39(1–3):55–72, 1978.

[89] Anna Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2002.

[90] Anna Gál and Pavel Pudlák. Monotone complexity and the rank of matrices. *Inform. Process. Lett.*, 87:321–326, 2003.

[91] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *45th STOC*, pages 467–476, 2013.

[92] Sanjam Garg, Abhishek Jain, Pratyay Mukherjee, Rohit Sinha, Mingyuan Wang, and Yinuo Zhang. Cryptography with weights: MPC, encryption and signatures. In *CRYPTO 2023*, volume 14081 of *LNCS*, page 295–327, 2023.

[93] Romain Gay, Iordanis Kerenidis, and Hoeteck Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In *CRYPTO 2015*, volume 9216 of *LNCS*, pages 485–502, 2015.

[94] Rosario Gennaro, Michael O. Rabin, and Tal Rabin. Simplified VSS and fact-track multiparty computations with applications to threshold cryptography. In *17th PODC*, pages 101–111, 1998.

[95] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *JCSS*, 60(3):592–629, 2000.

[96] M. Gharahi. *On the Complexity of Perfect Secret Sharing Schemes (in Persian)*. PhD thesis, Iran Univ. of Science and Technology, 2013.

[97] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *19th STOC*, pages 218–229, 1987.

[98] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. of the ACM*, pages 691–729, 1991.

[99] Jovan Dj. Golic. On matroid characterization of ideal secret sharing schemes. *J. Cryptol.*, 11(2):75–86, 1998.

[100] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *50th STOC*, pages 685–698, 2018.

[101] Viput Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *13th CCS*, pages 89–98, 2006.

[102] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20:71–86, 2000.

[103] Emirhan Gürpinar and Andrei E. Romashchenko. How to use undiscovered information inequalities: Direct applications of the copy lemma. In *ISIT 2019*, pages 1377–1381, 2019.

[104] Johan Håstad, Russel Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. on Computing*, 28(4):1364–1396, 1999.

[105] Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *63rd FOCS*, pages 968–979, 2022.

[106] Martin Hirt and Ueli Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. of Cryptology*, 13(1):31–60, 2000.

[107] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structure. In *Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1), 15-20, 1993.

[108] Wen-Ai Jackson and Keith M. Martin. Perfect secret sharing schemes on five participants. *Designs, Codes and Cryptography*, 9:267–286, 1996.

[109] Stasys Jukna. *Boolean Function Complexity – Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[110] Reza Kaboli, Shahram Khazaei, and Maghsoud Parviz. On ideal and weakly-ideal access structures. *Advances in Mathematics of Communications*, 17(3):697–713, 2021.

[111] Mauricio Karchmer and Avi Wigderson. On span programs. In *8th Structure in Complexity Theory*, pages 102–111, 1993.

[112] Ehud D. Karnin, Jonathan W. Greene, and Martin E. Hellman. On secret sharing systems. *IEEE Trans. on Information Theory*, 29(1):35–41, 1983.

[113] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *CRYPTO 2017*, volume 10401 of *Lecture Notes in Computer Science*, pages 357–388. Springer, 2017.

[114] Joe Kilian and Noam Nisan. Private communication, 1990.

[115] Ohad Klein and Ilan Komargodski. New bounds on the local leakage resilience of shamir's secret sharing scheme. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023*, volume 14081 of *LNCS*, pages 139–170. Springer, 2023.

[116] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. *J. Cryptol.*, 30(2):444–469, 2017.

[117] Hugo Krawczyk. Secret sharing made short. In *CRYPTO '93*, volume 773 of *LNCS*, pages 136–146, 1994.

[118] Stephan Krenn and Thomas Lorünser. *An Introduction to Secret Sharing*. Springer Cham, 2023.

[119] Kaoru Kurosawa, Koji Okada, Keiichi Sakano, Wakaha Ogata, and Shigeo Tsujii. Nonperfect secret sharing schemes and matroids. In *EUROCRYPT '93*, volume 765 of *LNCS*, pages 126–141, 1994.

[120] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[121] Kasper Green Larsen and Mark Simkin. Secret sharing lower bound: Either reconstruction is hard or shares are long. In *SCN 2020*, volume 12238 of *LNCS*, pages 566–578, 2020.

[122] Yehuda Lindell. Secure multiparty computation (MPC). Cryptology ePrint Archive, Paper 2020/300, 2020. `https://eprint.iacr.org/2020/300`.

[123] Tianren Liu and Vinod Vaikuntanathan. Breaking the circuit-size barrier in secret sharing. In *50th STOC*, pages 699–708, 2018.

[124] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In *CRYPTO 2017*, volume 10401 of *LNCS*, pages 758–790, 2017.

[125] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 567–596, 2018.

[126] Konstantin Makarychev, Yury Makarychev, Andrei E. Romashchenko, and Nikolai K. Vereshchagin. A new class of non-shannon-type inequalities for entropies. *Commun. Inf. Syst.*, 2(2):147–166, 2002.

[127] Jaume Martí-Farré and Carles Padró. Secret sharing schemes with three or four minimal qualified subsets. *Designs, Codes and Cryptography*, 34(1):17–34, 2005.

[128] Jaume Martí-Farré and Carles Padró. On secret sharing schemes, matroids and polymatroids. *J. Mathematical Cryptology*, 4(2):95–120, 2010.

[129] Jaume Martí-Farré, Carles Padró, and Leonor Vázquez. Optimal complexity of secret sharing schemes with four minimal qualified subsets. *Des. Codes Cryptogr.*, 61(2):167–186, 2011.

[130] Keith M. Martin, Maura B. Paterson, and Douglas R. Stinson. Error decodable secret sharing and one-round perfectly secure message transmission for general adversary structures. *Cryptography and Communications*, pages 65–86, 2011.

[131] Sebastià Martín, Carles Padró, and An Yang. Secret sharing, rank inequalities, and information inequalities. *IEEE Trans. Inf. Theory*, 62(1):599–609, 2016.

[132] Frantisek Matúš. Probabilistic conditional independence structures and matroid theory: Background. *Int. J. of General Systems*, 22:185–196, 1995.

[133] Frantisek Matúš. Matroid representations by partitions. *Discrete Mathematics*, 203:169–194, 1999.

[134] Frantisek Matúš. Infinitely many information inequalities. In *ISIT 2007*, pages 41–44, 2007.

[135] Frantisek Matúš. Two constructions on limits of entropy functions. *IEEE Trans. on Information Theory*, 53(1):320–330, 2007.

[136] Jessica Ruth Metcalf-Burton. Improved upper bounds for the information rates of the secret sharing schemes induced by the vámos matroid. *Discret. Math.*, 311(8-9):651–662, 2011.

[137] Paz Morillo, Carles Padró, Germán Sáez, and Jorge L. Villar. Weighted threshold secret sharing schemes. *Inform. Process. Lett.*, 70(5):211–216, 1999.

[138] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7:101–104, 1987.

[139] Moni Naor. Bit commitment using pseudorandom generators. *J. of Cryptology*, 4:151–158, 1991.

[140] Moni Naor and Avishai Wool. Access control and signatures via quorum secret sharing. *IEEE Transactions on Parallel and Distributed Systems*, 9(1):909–922, 1998.

[141] Peter Nelson. Almost all matroids are nonrepresentable. *Bulletin of the London Mathematical Society*, 50(2):245–248, 2018.

[142] Siaw-Lynn Ng. A representation of a family of secret sharing matroids. *Designs, Codes and Cryptography*, 30(1):5–19, 2003.

[143] Siaw-Lynn Ng and Michael Walker. On the composition of matroids and ideal secret sharing schemes. *Designs, Codes and Cryptography*, 24(1):49 – 67, 2001.

[144] James G. Oxley. *Matroid Theory*. Oxford University Press, 1992.

[145] Carles Padro. Lecture notes in secret sharing. Cryptology ePrint Archive, Paper 2012/674, 2013. `https://eprint.iacr.org/2012/674`.

[146] Carles Padró and Germán Sáez. Secret sharing schemes with bipartite access structure. *IEEE Trans. on Information Theory*, 46:2596–2605, 2000.

[147] Carles Padró, Leonor Vázquez, and An Yang. Finding lower bounds on the complexity of secret sharing schemes by linear programming. *Discret. Appl. Math.*, 161(7-8):1072–1084, 2013.

[148] Rudi Pendavingh and Stefan H. M. van Zwam. Skew partial fields, multilinear representations of matroids, and a matrix tree theorem. *Advances in Applied Mathematics*, 50(1):201 – 227, 2013.

[149] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *49th STOC*, pages 1246–1255, 2017.

[150] Toniann Pitassi and Robert Robere. Lifting Nullstellensatz to monotone span programs over any field. In *50th STOC*, pages 1207–1219, 2018.

[151] Michael O. Rabin. Randomized Byzantine generals. In *24th FOCS*, pages 403–409, 1983.

[152] Michael O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. of the ACM*, 36(2):335–348, 1989.

[153] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *21st STOC*, pages 73–85, 1989.

[154] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. In *38th FOCS*, pages 234–243, 1997.

[155] Alexander A. Razborov. A lower bound on the monotone network complexity of the logical permanent. *Mat. Zametki*, 37(6):887–900, 1985. In Russian, English translation in: *Math. Notes*, 37:485–493, 1985.

[156] Alexander A. Razborov. Lower bounds on monotone complexity of some Boolean functions. *Dokl. Ak. Nauk. SSSR*, 281:798–801, 1985. In Russian, English translation in: *Sov. Math. Dokl.*, 31:354–357, 1985.

[157] Robert Robere. *Unified Lower Bounds For Monotone Computation*. PhD thesis, University of Toronto, 2018. `https://www.cs.mcgill.ca/~robere/thesis.pdf`.

[158] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In *57th FOCS*, pages 406–415, 2016.

[159] Lajos Rónyai, László Babai, and Murali K. Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the AMS*, 14(3):717–735, 2001.

[160] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT 2005*, pages 457–473, 2005.

[161] Paul D. Seymour. A forbidden minor characterization of matroid ports. *Quart. J. Math. Oxford Ser.*, 27:407–413, 1976.

[162] Paul D. Seymour. On secret-sharing matroids. *J. of Combinatorial Theory, Series B*, 56:69–73, 1992.

[163] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[164] Bhavani Shankar, Kannan Srinathan, and C. Pandu Rangan. Alternative protocols for generalized oblivious transfer. In *9th ICDCN*, volume 4904 of *LNCS*, pages 304–309, 2008.

[165] Gustavus J. Simmons. How to (really) share a secret. In *CRYPTO '88*, volume 403 of *LNCS*, pages 390–448, 1990.

[166] Gustavus J. Simmons, Wen-Ai Jackson, and Keith M. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.

[167] Juriaan Simonis and Alexei Ashikhmin. Almost affine codes. *Designs, Codes and Cryptography*, 14(2):179–197, 1998.

[168] Douglas R. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2:357–390, 1992.

[169] Douglas R. Stinson and Ruizhong Wei. An application of ramp schemes to broadcast encryption. *Inform. Process. Lett.*, pages 131–135, 1999.

[170] Hung-Min Sun and Shiuh-Pyng Shieh. Secret sharing in graph-based prohibited structures. In *INFOCOM '97*, pages 718–724. IEEE, 1997.

[171] Eva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

[172] Tamir Tassa. Hierarchical threshold secret sharing. In *TCC 2004*, volume 2951 of *LNCS*, pages 473–490, 2004.

[173] Tamir Tassa. Generalized oblivious transfer by secret sharing. *Designs, Codes and Cryptography*, 58(1):11–21, 2011.

[174] Tamir Tassa and Nira Dyn. Multipartite secret sharing by bivariate interpolation. In *33rd ICALP*, volume 4052 of *LNCS*, pages 288–299, 2006.

[175] Martin Tompa and Heather Woll. How to share a secret with cheaters. *J. Cryptol.*, 1(2):133–138, 1988.

[176] Andrei Tonkikh and Luciano Freitas de Souza. Swiper: A new paradigm for efficient weighted distributed protocols. In Ran Gelles, Dennis Olivetti, and Petr Kuznetsov, editors, *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing, PODC 2024*, pages 283–294. ACM, 2024.

[177] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *CRYPTO 2022*, volume 13507 of *LNCS*, pages 535–559, 2022.

[178] Vinod Vaikuntanathan and Prashant Nalini Vasudevan. Secret sharing and statistical zero knowledge. In *ASIACRYPT 2015*, pages 656–680, 2015.

[179] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-io from evasive LWE. In *ASIACRYPT 2022*, volume 13791 of *LNCS*, pages 195–221, 2022.

[180] V. Vinod, Arvind Narayanan, K.Srinathan, C. Pandu Rangan, and Kwangjo Kim. On the power of computational secret sharing. In *Indocrypt 2003*, volume 2904 of *LNCS*, pages 162–176, 2003.

[181] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *PKC 2011*, volume 6571 of *LNCS*, pages 53–70, 2011.

[182] Hoeteck Wee. Dual system encryption via predicate encodings. In *TCC 2014*, volume 8349 of *LNCS*, pages 616–637, 2014.

[183] Dominic J. A. Welsh. *Matroid Theory*. Academic press, London, 1976.

[184] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

[185] Andrew Chi-Chih Yao. Protocols for secure computations. In *23th FOCS*, pages 160–164, 1982.

[186] Andrew Chi-Chih Yao. Unpublished manuscript, 1989. Presented at Oberwolfach and DIMACS workshops.

[187] Raymond W. Yeung. *Information Theory and Network Coding*. Springer, 2008.

[188] Zhen Zhang and Raymond W. Yeung. On characterization of entropy function via information inequalities. *IEEE Trans. on Information Theory*, 44(4):1440–1452, 1998.

Appendix A

Background on Complexity, Cryptography, and Information Theory

A.1 Background in Complexity

We recall the definition of monotone formulas and monotone circuits. A monotone Boolean circuit C with n inputs is a labeled directed acyclic graph G, with one vertex with out-degree 0 (the root); each vertex with in-degree 0 (i.e., a leaf) is labeled by some variable x_i[29] and each internal node is labeled by either $\wedge$ or by $\vee$. For monotone circuits, we can assume, w.l.o.g., that for each variable x_i there is a unique leaf labeled by x_i. We will sometimes allow the internal nodes to be labeled by other functions. We consider a circuit in which the in-degree of each node can be arbitrary. An example of a monotone circuit is given in Figure A.1.

For an assignment $y = (y_1, \ldots, y_n) \in \{0,1\}^n$, the value $C(y)$ is computed as follows: We sort the nodes in the graph G in a topological order, where if there is an edge from u to v, then u will appear before v in the sorted list. We compute the values of the nodes (also known as gates) in the circuits according to this order:

v is a leaf labeled by variable x_i. The value of v is y_i,

v is an internal node label by $\wedge$ whose in-coming neighbors are $u_1, \ldots, u_\ell$. Let $o_1, \ldots, o_\ell$ be the values computed for $u_1, \ldots, u_\ell$ (since for $1 \leq i \leq \ell$ there is an edge (u_i, v), these values have already been computed). The value of v is $o_1 \wedge o_1 \wedge \cdots \wedge o_\ell$.

v is an internal node label by $\vee$ whose in-coming neighbors are $u_1, \ldots, u_\ell$. Let $o_1, \ldots, o_\ell$ be the values computed for $u_1, \ldots, u_\ell$. The value of v is $o_1 \vee o_1 \vee \cdots \vee o_\ell$.

The value $C(y)$ is the value of the root. A monotone *formula* is a monotone circuit in which the out-degree of each node, except for the root, is 1, i.e., the graph G is a directed tree (where edges are directed towards the root). The size of a circuit/formula is the number of nodes in G. Every monotone circuit computes a monotone function and every monotone function can be computed by a monotone formula. However, for

[29] We can also consider non-monotone circuits, where a leaf can be labeled by a negated variable $\overline{x_i}$.

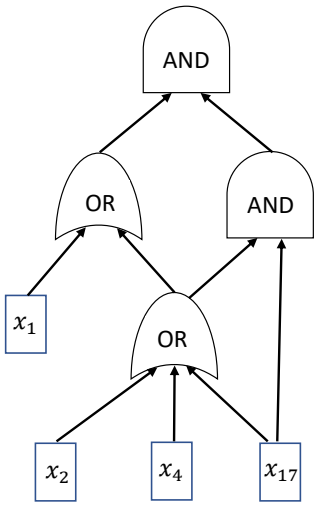


Figure A.1: An example of a monotone circuit. For example, the fan-in of the bottom OR gate is 3 and its fan-out is 2.

almost all monotone functions $f : \{0,1\}^n \to \{0,1\}$ the size of the smallest monotone circuits computing them is $2^{\Omega(n)}$. It is known that for some explicit function f the size of the smallest monotone formula for f is exponentially bigger than the size of the smallest monotone circuit for f (e.g., [154]).

An arithmetic circuit is similar to Boolean circuits; however, the gates in it are addition and multiplication. Formally, an arithmetic circuit over $\mathbb{F}$ with n inputs is an acyclic graph where:

- There is a unique node with out-degree 0. This node is called the output node.
- There are n nodes with in-degree 0, (i.e., leaves) called input nodes. For each i, where $1 \leq i \leq n$, there is a node labeled by the variable x_i.[30]
- Each non-leaf is labeled either by $\times$, called a multiplication gate, or by $+$, called an addition gate.

The size of the circuit is the number of nodes in the circuit. For technical reasons, we will assume in Chapter 6 that the fan-in of every non-leaf is two. Every arithmetic circuit of size s can be transformed into a circuit with fan-in 2 and size $O(s^2)$. The function computed by an arithmetic circuit over a field $\mathbb{F}$ is defined in the natural way, where the arithmetic is done over $\mathbb{F}$. Every function $f : \mathbb{F}^n \to \mathbb{F}$ can be represented by an arithmetic circuit (however, the size of the circuit might be exponential in n). In particular, every Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be represented by an arithmetic circuit over any field $\mathbb{F}$. By working over an extension field, we can assume that $|\mathbb{F}| > n$.

[30]There can be additional nodes with in-degree 0 labeled by constants. For simplicity, we ignore such nodes.

A.2 Background in Cryptography

We say that a function $\mathrm{negl}(\lambda)$ is negligible if it is smaller than $1/p(\lambda)$ for every positive polynomial; formally, for every positive polynomial $p(\lambda)$ there exists an integer λ_0 such that $\mathrm{negl}(\lambda) \leq 1/p(\lambda)$ for every $\lambda \geq \lambda_0$.

As is common in cryptography, we consider a non-uniform polynomial-time adversary that when given an input x of length n also gets a polynomially-long advice string h_n (which only depends on the length of the input) and tries to break a system. The system is secure if for every sequence of advice strings, the adversary cannot break the system. Formally, a non-uniform polynomial-time algorithm is a polynomial-time algorithm ALG and a sequence of advice strings $(h_n)_{n\in\mathbb{N}}$ such that:

- The sequence is polynomially bounded, that is, there exists a constant c for which $|h_n| \leq n^c$ for every $n \in \mathbb{N}$,
- For every $n \in \mathbb{N}$ and every $x \in \{0,1\}^n$, the algorithm outputs $\mathrm{ALG}(x, h_n)$.

We next provide the definition of a non-interactive commitment scheme with perfect binding.

Definition A.1. *A commitment scheme is a randomized polynomial-time algorithm* Commit*, whose inputs are a security parameter 1^λ (in unary), a message m, and a random string r; its output is a string $c \leftarrow \mathsf{Commit}(1^\lambda, m; r)$. A commitment scheme with perfect binding should satisfy the following two requirements:*

Computational hiding. *Consider the following game between a committer and an adversary:*

- *The adversary with input a security parameter 1^λ chooses two messages m_0, m_1 such that $|m_0| = |m_1|$ and sends them to the committer.*
- *The committer chooses a uniformly distributed bit $b \in \{0,1\}$ and a uniformly distributed r, computes $c \leftarrow \mathsf{Commit}(1^\lambda, m_b; r)$, and sends c to the adversary.*
- *The adversary outputs a bit b' and wins if $b = b'$.*

The commitment scheme Commit *is hiding if for every non-uniform polynomial-time adversary $\mathcal{A}$ there exists a negligible function* $\mathrm{negl}(\lambda)$ *such that the probability that $\mathcal{A}$ wins is at most* $1/2 + \mathrm{negl}(\lambda)$.

Perfect binding. *For any $m_0 \neq m_1$ such that $|m_0| = |m_1|$ and r_0, r_1:*

$$\mathsf{Commit}(1^\lambda, m_0; r_0) \neq \mathsf{Commit}(1^\lambda, m_1; r_1).$$

A.3 The Entropy Function and Its properties

In this appendix we provide the definition of entropy and conditional entropy and discuss some properties of these quantities. For more background on the entropy and for proofs of the properties, the reader may consult, e.g., [63].

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

$$H(X) \stackrel{\text{def}}{=} \sum_{x \in \text{SUPPORT}(X)} \Pr[X = x] \log\left(\frac{1}{\Pr[X = x]}\right), \tag{A.1}$$

where if there is an $x \in X$ such that $\Pr[X = x] = 1$ then $H(X) \stackrel{\text{def}}{=} 1$. It holds that

$$0 \leq H(X) \leq \log(|\text{SUPPORT}(X)|). \tag{A.2}$$

Intuitively, $H(X)$ measures the amount of uncertainty in X where $H(X) = 0$ if X is deterministic, i.e., there is a value x such that $\Pr[X = x] = 1$, and $H(X) = \log(|\text{SUPPORT}(X)|)$ if X is uniformly distributed over $\text{SUPPORT}(X)$. The concatenation of two variables X, Y is denoted by XY or X, Y. Given three jointly distributed random variables X, Y, and Z define the *conditional entropy* as

$$H(X|YZ) \stackrel{\text{def}}{=} H(XY|Z) - H(Y|Z). \tag{A.3}$$

The conditional entropy is non-negative and conditioning on a variable only decreases the uncertainty on X, that is for every X, Y, Z

$$0 \leq H(X|YZ) \leq H(X|Y) \leq H(X). \tag{A.4}$$

If Z is a deterministic function of Y, i.e., $H(Z|Y) = 0$, then conditioning on YZ is equivalent to conditioning on Z, that is,

$$\text{If } H(Z|Y) = 0 \text{ then } H(X|YZ) = H(X|Y). \tag{A.5}$$

The entropy is subadditive, that is, the (conditional) joint entropy of two random variables is at most the sum of the entropies of each variable.

$$H(X|Z) \leq H(XY|Z) = H(X|Z) + H(Y|XZ) \leq H(X|Z) + H(Y|Z). \tag{A.6}$$

Two random variables X and Y are independent iff $H(X|Y) = H(X)$ and the value of Y implies the value of X iff $H(X|Y) = 0$.

Definition A.2 (Statistical distance)**.** *The* statistical distance *between two random variables A and B is the function*

$$\text{SD}(A, B) = \frac{1}{2} \sum_{\alpha \in \text{SUPPORT}(A) \cup \text{SUPPORT}(B)} \Big| \Pr[A = \alpha] - \Pr[B = \alpha] \Big|.$$