Secret-Sharing Schemes: A Survey

Amos Beimel

Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel.

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 important tools in cryptography and they are used as a building box in many secure protocols, e.g., general protocol for multiparty computation, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, and generalized oblivious transfer.
In this survey, we will describe the most important constructions of secret-sharing schemes, explaining the connections between secret-sharing schemes and monotone formulae and monotone span programs. The main problem with known secret-sharing schemes is the large share size: it is exponential in the number of parties. We conjecture that this is unavoidable. We will discuss 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 is a class of schemes based on linear algebra that contains most known schemes, super-polynomial lower bounds on the share size are known. We will describe the proofs of these lower bounds. We will also present two results connecting secret-sharing schemes for a Hamiltonian access structure to the NP vs. coNP problem and to a major open problem in cryptography – constructing oblivious-transfer protocols from one-way functions.

1 Introduction

Secret-sharing schemes are a tool used in many cryptographic protocols. A secret-sharing scheme involves a dealer who has a secret, a set of n parties, and a collection $\mathcal{A}$ of subsets of parties called the access structure. A secret-sharing scheme for $\mathcal{A}$ is a method by which the dealer distributes shares to the parties such that: (1) any subset in $\mathcal{A}$ can reconstruct the secret from its shares, and (2) any subset not in $\mathcal{A}$ cannot reveal any partial information on the secret. Originally motivated by the problem of secure information storage, secret-sharing schemes have found numerous other applications in cryptography and distributed computing, e.g., Byzantine agreement [54], secure multiparty computations [13, 24, 28], threshold cryptography [31], access control [52], attribute-based encryption [40, 68], and generalized oblivious transfer [59, 65].

Example 1 (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 user's credentials. In an attributed-based encryption system, presented by Sahai and Waters [57], 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 [40, 68], 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.

Secret-sharing schemes were introduced by Blakley [17] and Shamir [58] for the threshold case, that is, for the case where the subsets that can reconstruct the secret are all the sets whose cardinality is at least a certain threshold. Secret-sharing schemes for general access structures were introduced and constructed by Ito, Saito, and Nishizeki [45]. More efficient schemes were presented in, e.g., [14, 61, 21, 46, 16]. Specifically, Benaloh and Leichter [14] proved that if an access structure can be described by a small *monotone formula* then it has an efficient perfect secret-sharing scheme. This was generalized by Karchmer and Wigderson [46] who showed that if an access structure can be described by a small *monotone span program* then it has an efficient scheme (a special case of this construction appeared before in [21]).

A major problem with secret-sharing schemes is that the shares' size in the best known secret-sharing schemes realizing general access structures is exponential in the number of parties in the access structure. Thus, the known constructions for general access structures are impractical. This is true even for explicit access structures (e.g., access structures whose characteristic function can be computed by a small uniform circuit). On the other hand, the best known lower bounds on the shares' size for sharing a secret with respect to an access structure (e.g., in [23, 29]) are far from the above upper bounds. The best lower bound was proved by Csirmaz [29], proving that, for every n, there is an access structure with n parties such that sharing ℓ-bit secrets requires shares of length $\Omega(\ell n/\log n)$. The question if there exist more efficient schemes, or if there exists an access structure that does not have (space) efficient schemes remains open. The following is a widely believed conjecture (see, e.g., [3]):

Conjecture 1. There exists an $\epsilon > 0$ such that for every integer n there is an access structure with n parties, for which every secret-sharing scheme distributes shares of length exponential in the number of parties, that is, $2^{\epsilon n}$.

Proving (or disproving) this conjecture is one of the most important open questions concerning secret sharing. No major progress on proving or disproving this conjecture has been obtained in the last 16 years. It is not known how to prove that there exists an access structure that requires super-polynomial shares (even for an implicit access structure).

Most previously known secret-sharing schemes are *linear*. 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 [58, 17, 45, 14, 61, 16, 46] are all linear. For

many application, the linearity is important, e.g., for secure multiparty computation as will be described in Section 4. Thus, studying linear secret-sharing schemes and their limitations is important. Linear secret-sharing schemes are equivalent to monotone span programs, defined by [46]. Super-polynomial lower bounds for monotone span programs and, therefore, for linear secret-sharing schemes were proved in [5, 2, 36].

In this survey we will present two unpublished results of Rudich [56]. Rudich considered a Hamiltonian access structure, the parties in this access structure are edges in a complete undirected graph, and a set of edges (parties) is authorized if it contains a Hamiltonian cycle.[1] Rudich proved that if $NP \neq coNP$, then this access structure does not have a secret-sharing scheme in which the sharing of the secret can be done by a polynomial-time algorithm. As efficient sharing of secrets is essential in applications of secret-sharing, Rudich's results implies that there is no practical scheme for the Hamiltonian access structure. Furthermore, Rudich proved that if one-way functions exist and if the Hamiltonian access structure has a computational secret-sharing scheme (with efficient sharing and reconstruction), then efficient protocols for oblivious transfer exists. Thus, constructing a computational secret-sharing scheme for the Hamiltonian access structure will solve a major open problem in cryptography, i.e., using Impagliazzo's terminology [43], it will prove that Minicrypt = Cryptomania.

1.1 Organization

This survey is intended for readers with some background in cryptography and complexity. When possible, we try to give the required definitions. The rest of the survey is organized as follows. In Section 2 we define secret-sharing schemes, giving two definitions and proving that they are equivalent. In Section 3, we present constructions of secret-sharing schemes. In Section 4, we show how to construct secure multiparty protocols for general functions (in the honest-but-curious model) using secret-sharing schemes. In Section 5, we discuss lower bounds for secret-sharing schemes and present the best known lower bounds for general secret-sharing schemes and linear secret-sharing schemes. In Section 6, we present the unpublished results of Rudich. Finally, in Section 7, we summarize this survey and mention the most important open problems for secret sharing.

2 Definitions

In this section we define secret-sharing schemes. We supply two definitions and argue that they are equivalent.

Definition 1 (Access Structure, Distribution Scheme). *Let* $\{p_1, \ldots, p_n\}$ *be a set of parties. A collection* $\mathcal{A} \subseteq 2^{\{p_1,\ldots,p_n\}}$ *is* monotone *if* $B \in \mathcal{A}$ *and* $B \subseteq C$ *imply that* $C \in \mathcal{A}$. *An* access structure *is a monotone collection* $\mathcal{A} \subseteq 2^{\{p_1,\ldots,p_n\}}$

[1] The results of Rudich apply to other monotone NP-complete problem as well, e.g., the clique problem.

of non-empty subsets of $\{p_1, \ldots, p_n\}$*. Sets in* $\mathcal{A}$ *are called* authorized*, and sets not in* $\mathcal{A}$ *are called* unauthorized*.*

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

The information ratio *of a distribution scheme is* $\frac{\max_{1 \leq j \leq n} \log |K_j|}{\log |K|}$*. The* average information ratio *of a distribution scheme is* $\frac{\sum_{1 \leq j \leq n} \log |K_j|}{n \cdot \log |K|}$*.* [2]

We start with a definition of secret-sharing as given in [26, 4, 12].

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

Correctness. *The secret* k *can be reconstructed by any authorized set of parties. That is, for any set* $B \in \mathcal{A}$ *(where* $B = \{p_{i_1}, \ldots, p_{i_{|B|}}\}$*), there exists a* reconstruction function $\textsc{Recon}_B : K_{i_1} \times \ldots \times K_{i_{|B|}} \to K$ *such that for every* $k \in K$*,*

$$\Pr[\, \textsc{Recon}_B(\Pi(k, r)_B) = k \,] = 1. \tag{1}$$

Perfect Privacy. *Every unauthorized set cannot learn anything about the secret (in the information theoretic sense) from their shares. Formally, for any set* $T \notin \mathcal{A}$*, for every two secrets* $a, b \in K$*, and for every possible vector of shares* $\langle s_j \rangle_{p_j \in T}$*:*

$$\Pr[\, \Pi(a, r)_T = \langle s_j \rangle_{p_j \in T} \,] = \Pr[\, \Pi(b, r)_T = \langle s_j \rangle_{p_j \in T} \,]. \tag{2}$$

Remark 1. In the above definition, we required correctness with probability 1 and perfect privacy: for every two secrets a, b the distributions $\Pi(a, r)_T$ and $\Pi(a, r)_T$ are identical. We can relax these requirements and require that the correctness holds with high probability and that the statistical distance between $\Pi(a, r)_T$ and $\Pi(a, r)_T$ is small. Schemes that satisfy these relaxed requirements are called statistical secret-sharing schemes. For example, such schemes are designed in [16].

We next define an alternative definition of secret-sharing schemes originating in [47, 23]; this definition uses the entropy function. For this definition we assume that there is some known probability distribution on the domain of secrets K.

[2] In the secret sharing literature it is also common to use the term *information rate*, which is the inverse of the information ratio.

Any probability distribution on the domain of secrets, together with the distribution scheme Σ, 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 S_A, and by S the random variable denoting the secret. The privacy in the alternative definition requires that if $T \notin \mathcal{A}$, then the random variables S and S_T are independent.

As traditional in the secret sharing literature, we formalize the above two requirements using the entropy function. The support of a random variables 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 $H(X) \stackrel{\text{def}}{=} \sum \Pr[X = x] \log 1/\Pr[X = x]$, where the sum is taken over all values x in the support of X, i.e., all values x such that $\Pr[X = x] > 0$. It holds that $0 \leq H(X) \leq \log |\text{SUPPORT}(X)|$. 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 $\text{SUPPORT}(X)$. Given two random variables X and Y we consider their concatenation XY and define the *conditional entropy* as $H(X|Y) \stackrel{\text{def}}{=} H(XY) - H(Y)$. It holds that $0 \leq H(X|Y) \leq H(X)$; 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 $H(X|Y) = 0$. For more background on the entropy function, the reader may consult [27].

Definition 3 (Secret Sharing – Alternative Definition). *We say that a distribution scheme is a* secret-sharing scheme realizing an access structure $\mathcal{A}$ with respect to a given probability distribution on the secrets, denoted by a random variable S, if the following conditions hold.

CORRECTNESS. For every authorized set $B \in \mathcal{A}$,

$$H(S|S_B) = 0. \tag{3}$$

PRIVACY. For every unauthorized set $T \notin \mathcal{A}$,

$$H(S|S_T) = H(S). \tag{4}$$

Definition 2 and Definition 3 are equivalent, as proved below in Claim 1. The advantage of Definition 2 is that it does not assume that there is a probability distribution on the secrets and that this distribution is known. Furthermore, Definition 2 can be generalized to statistical secret sharing and computational secret sharing. On the other hand, Definition 3 is more convenient for proving lower bounds. Thus, the equivalence of the definitions allows choosing the more suitable definition for the specific task.

Furthermore, the equivalence of the definitions allows proving a result of Blundo et al. [20] that the privacy of a scheme according to Definition 3 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.

Claim 1. *The following claims are equivalent for a distribution scheme Σ:*

1. *The scheme Σ is secure according to Definition 2.*
2. *There is some distribution on the secrets with support K (that is, $\Pr[S = a] > 0$ for every $a \in K$) such that the scheme is secure according to Definition 3 with respect to this distribution.*
3. *For every distribution on the secrets whose support is contained in K, the scheme is secure according to Definition 3 with respect to the distribution.*

Proof. We first show that (1) implies (3) (and, hence, (2)). Let $\Sigma = \langle \Pi, \mu \rangle$ be a secret-sharing scheme which is private according to Definition 2, and let S be random variable distributed according to some distribution over K. Thus, for any set $T \notin \mathcal{A}$, any secret $a \in K$, and any shares $\langle s_j \rangle_{p_j \in T}$ for the parties in T,

$$\begin{aligned}
\Pr[\, S_T = \langle s_j \rangle_{p_j \in T} | S = a \,] &= \Pr[\, \Pi(a, r)_T = \langle s_j \rangle_{p_j \in T} \,] \\
&= \sum_{b \in K} \Pr[S = b] \cdot \Pr[\, \Pi(b, r)_T = \langle s_j \rangle_{p_j \in T} \,] \qquad (5) \\
&= \sum_{b \in K} \Pr[S = b] \cdot \Pr[\, S_T = \langle s_j \rangle_{p_j \in T} | S = b] \,] \\
&= \Pr[\, S_T = \langle s_j \rangle_{p_j \in T} \,], \qquad (6)
\end{aligned}$$

where the equality in (5) follows from (2). In other words, by (6), 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 private according to Definition 3 with respect to this distribution on S.

Now assume that $\Sigma = \langle \Pi, \mu \rangle$ is a secret-sharing scheme which is private according to Definition 3 for some fixed distribution on the secrets with support K, that is, assume that (2) holds. For any set $T \notin \mathcal{A}$, the random variables S_T and S are independent, and, in particular, for every pair of secrets $a, b \in K$, and every shares $\langle s_j \rangle_{p_j \in T}$

$$\Pr_r[\, \Pi(a, r)_T = \langle s_j \rangle_{p_j \in T} \,] = \Pr_{r,k}[\, \Pi(k, r)_T = \langle s_j \rangle_{p_j \in T} \,] = \Pr_r[\, \Pi(b, r)_T = \langle s_j \rangle_{p_j \in T} \,],$$

where the first and last probabilities are for fixed secrets and are taken over the randomness of Π, and the middle probability is over both the randomness of Π and the secret k chosen according to the fixed distribution. Thus, the scheme is secure according to Definition 2. □

3 Constructions of Secret-Sharing Schemes

In this section we describe some of the most interesting constructions of secret-sharing schemes.

3.1 Shamir's Threshold Secret-Sharing Scheme

In a threshold secret-sharing schemes, the authorized sets are all sets whose size is bigger than some threshold, that is, they realize the t-out-of-n access structure $\mathcal{A}_t = \{A \subseteq \{p_1, \ldots, p_n\} : |A| \geq t\}$, where $1 \leq t \leq n$ is an integer. Shamir [58] 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 $q > n$. 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 $\alpha_j = j$). To share a secret $k \in \mathbb{F}_q$, the dealer chooses $t-1$ 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 $P(x) = k + \sum_{i=1}^{t} a_i x^i$. The share of p_j is $s_j = P(\alpha_j)$ (where P is evaluated using the arithmetic of $\mathbb{F}_q$).

The correctness and privacy of Shamir's scheme follow from the Lagrange's interpolation theorem: For every field $\mathbb{F}$, every t distinct values $x_1, \ldots, x_t$, 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 $1 \leq j \leq t$.

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

$$Q(x) = \sum_{\ell=1}^{t} s_{i_\ell} \prod_{1 \leq j \leq t, j \neq \ell} \frac{\alpha_{i_j} - x}{\alpha_{i_j} - \alpha_{i_\ell}}.$$

Notice that $Q(\alpha_{i_\ell}) = s_{i_\ell} = P(\alpha_{i_\ell})$ for $1 \leq \ell \leq t$. That is, P and Q are polynomial of degree at most $t-1$ that agree on t points, thus, by the uniqueness in the interpolation theorem, P and Q are equal, and, in particular, $Q(0) = P(0) = k$. Thus, the parties in B reconstruct k by computing

$$k = Q(0) = \sum_{\ell=1}^{t} s_{i_\ell} \prod_{1 \leq j \leq t, j \neq \ell} \frac{\alpha_{i_j}}{\alpha_{i_j} - \alpha_{i_\ell}}.$$

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

$$k = \sum_{\ell=1}^{t} \beta_\ell \cdot s_{i_\ell}, \text{ where } \beta_\ell = \prod_{1 \leq j \leq t, j \neq \ell} \frac{\alpha_{i_j}}{\alpha_{i_j} - \alpha_{i_\ell}}. \tag{7}$$

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

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 $T = \{p_{i_1}, \ldots, p_{i_{t-1}}\}$ and every $a \in \mathbb{F}_q$, there is a unique polynomial P_a with degree at most $t-1$ such that $P_a(0) = a$ and $P_a(\alpha_{i_\ell}) = s_{i_\ell}$ for $1 \leq \ell \leq t-1$. Hence,

$$\Pr[\, \Pi(a, r)_T = \langle s_{i_\ell} \rangle_{1 \leq \ell \leq t-1} \,] = \frac{1}{q^{t-1}}.$$

Since this probability is the same for every $a \in \mathbb{F}_q$, the privacy follows.

3.2 Undirected s-t-Connectivity

Consider the access structure $\mathcal{A}_{\text{ustcon}}$, 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 [15] constructed a secret-sharing scheme realizing this access structure. We next describe this secret-sharing scheme. Let $k \in \{0,1\}$ be a secret. To share k, the dealer chooses $m-2$ random bits $r_2, \ldots, r_{m-1}$ independently with uniform distribution. Furthermore, the dealer sets $r_1 = k$ and $r_m = 0$. The share of a party (v_i, v_j) is $r_i \oplus r_j$.[3] To see that this scheme is correct, consider a set of parties which is 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:

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

To see that this scheme is private 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 exist 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 $\{s_{i,j}\}_{(i,j)\in T}$ be a set of shares generated for the parties in T with the secret $k = 0$, where $s_{i,j}$ is the share given to the party (v_i, v_j). We next show that the number of vectors of random bits $r_1, r_2, \ldots, r_m$ that generate $\{s_{i,j}\}_{(i,j)\in T}$ given the secret $k = 0$ is equal to the number of vectors of random bits that generate these shares given the secret $k = 1$. Fix a vector of random bits $r_1, r_2, \ldots, r_{m-1}, r_m$ that generate the shares $\{s_{i,j}\}_{(i,j)\in T}$ with the secret $k = 0$. Recall that $r_1 = k = 0$ 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. Notice that $r'_1 = 1$ and $r'_m = 0$. Thus, these bits generate shares for the secret $k' = 1$. We claim that the random bits $r'_1, \ldots, r'_m$ generate the shares $\{s_{i,j}\}_{(i,j)\in T}$ with the secret $k = 1$. There are only two cases to consider:

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

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

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

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

[3] We can generalize this scheme such that the domain of secrets is any finite group H. To share a secret $k \in H$, the dealer chooses $m-2$ random elements $r_2, \ldots, r_{m-1}$ from H independently with uniform distribution, and sets $r_1 = k$ and $r_m = 0$. The share of a party (v_i, v_j), where $i < j$, is $r_j - r_i$.

To conclude, the number of vectors of random bits that generate the shares $\{s_{i,j}\}_{(i,j)\in T}$ 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 the scheme is private.

3.3 Ito, Saito, and Nishizeki's Constructions [45]

Ito, Saito, and Nishizeki [45] defined secret-sharing schemes for general access structures and showed how to construct such schemes for every monotone access structure. Specifically, let $\mathcal{A}$ be any monotone access structure. The dealer shares the secret independently for each authorized set $B \in \mathcal{A}$. That is, to share a secret $k \in \{0,1\}$, the dealer does the following for every authorized set $B \in \mathcal{A}$, where $B = \{p_{i_1}, \ldots, p_{i_\ell}\}$:

- chooses $\ell - 1$ random bits $r_1, \ldots, r_{\ell-1}$,
- computes $r_\ell = k \oplus r_1 \oplus \cdots \oplus r_{\ell-1}$, and
- gives p_{i_j} the bit r_j.

We emphasize that for each set $B \in \mathcal{A}$ the random bits are chosen by the dealer independently. Clearly, each set in $\mathcal{A}$ can reconstruct the secret by computing the exclusive-or of the bits given to the set. On the other hand, each unauthorized set $T \notin \mathcal{A}$ misses at least one party from each authorized set, thus, misses at least one bit given to the authorized set. In other words, the bits held by the parties in T are uniformly distributed and independent of the secret.

To summarize, the number of bits that p_j gets is the number of authorized sets that contain p_j. A simple optimization is to share the secret k only for minimal authorized sets. Still, this scheme is highly inefficient for access structures in which the number of minimal sets is big. For example, consider the $n/2$-out-of-n access structure, that is, the access structure

$$\mathcal{A}_{n/2} = \{B \subset \{p_1, \ldots, p_n\} \,:\, |B| \geq n/2\}\,.$$

The number of bits that each party gets in the scheme of [45] is $\binom{n-1}{n/2-1} = \Theta(2^n/\sqrt{n})$. On the other hand, Shamir's scheme for this access structure gives each party a share whose size is the same as the size of the secret.

3.4 The Monotone Formulae Construction [14]

Benaloh and Leichter [14] describe a construction of secret-sharing schemes for any access structure based on monotone formulae. The construction of [14] generalizes the construction of [45] and is more efficient. However, also in this scheme for most access structures the length of the shares is exponential in the number of parties even for a one-bit secret.

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. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two access structures. We assume that they have

the same set of parties $\{p_1, \ldots, p_n\}$. However, 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. We define two new access structures, where $B \in \mathcal{A}_1 \vee \mathcal{A}_2$ iff $B \in \mathcal{A}_1$ or $B \in \mathcal{A}_2$, and $B \in \mathcal{A}_1 \wedge \mathcal{A}_2$ iff $B \in \mathcal{A}_1$ and $B \in \mathcal{A}_2$. We assume that for $i \in \{1, 2\}$ there is a secret-sharing scheme Σ_i realizing $\mathcal{A}_i$, where the two schemes have same domain of secrets $K = \{0, \ldots, m-1\}$ for some $m \in \mathbb{N}$. Furthermore, assume that for every $1 \leq j \leq n$ the share of p_j in the scheme Σ_i is an element in $K^{a_{i,j}}$ for every $i \in \{1, 2\}$, and denote $a_j = a_{1,j} + a_{2,j}$. Then there exist secret-sharing schemes realizing $\mathcal{A}_1 \vee \mathcal{A}_2$ and $\mathcal{A}_1 \wedge \mathcal{A}_2$ in which the domain of shares of p_j is K^{a_j}:

- To share a secret $k \in K$ for the access structure $\mathcal{A}_1 \vee \mathcal{A}_2$, independently share k using the scheme Σ_i (realizing $\mathcal{A}_i$) for $i \in \{1, 2\}$.
- To share a secret $k \in K$ for the access structure $\mathcal{A}_1 \wedge \mathcal{A}_2$, choose $k_1 \in K$ with uniform distribution and let $k_2 = (k - k_1) \bmod m$. Next, for $i \in \{1, 2\}$, independently share k_i using the scheme Σ_i (realizing $\mathcal{A}_i$). For every set $B \in \mathcal{A}_1 \wedge \mathcal{A}_2$, the parties in B can reconstruct both k_1 and k_2 and compute $k = (k_1 + k_2) \bmod m$. On the other hand, for every set $T \notin \mathcal{A}$, the parties in T do not have any information on at least one k_i, hence do not have any information on the secret k.

For example, given an access structure $\mathcal{A} = \{B_1, \ldots, B_\ell\}$, we define $\mathcal{A}_i = \{B_1, \ldots, B_i\}$. Clearly, $\mathcal{A}_i = \mathcal{A}_{i-1} \vee \{B_i\}$, and for every $1 \leq i \leq \ell$ there is a scheme realizing $\{B_i\}$ with a domain of secrets $\{0, 1\}$, where each $p_j \in B$ gets a one-bit share. Applying the scheme of Benaloh and Leichter recursively, we get the scheme of Ito, Saito, and Nishizeki.

The scheme of Benaloh and Leichter can efficiently realize 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 the access structures that can be efficiently realized by Benaloh and Leichter's scheme it is convenient to view an access structure as a function. We describe each set $A \subseteq \{p_1, \ldots, p_n\}$ by its characteristic vector (string) $v_A \in \{0, 1\}^n$, where $v_A[j] = 1$ iff $p_j \in A$. With an access structure $\mathcal{A}$, we associate the function $f_{\mathcal{A}} : \{0, 1\}^n \to \{0, 1\}$, where $f_{\mathcal{A}}(v_B) = 1$ iff $B \in \mathcal{A}$. We say that $f_{\mathcal{A}}$ describes $\mathcal{A}$. As $\mathcal{A}$ is monotone, the function $f_{\mathcal{A}}$ is monotone. Furthermore, for two access structures $\mathcal{A}_1$ and $\mathcal{A}_2$ if $f_1 = f_{\mathcal{A}_1}$ and $f_2 = f_{\mathcal{A}_2}$, then $f_1 \vee f_2 = f_{\mathcal{A}_1 \vee \mathcal{A}_2}$ and $f_1 \wedge f_2 = f_{\mathcal{A}_1 \wedge \mathcal{A}_2}$. Using this observation, the scheme of Benaloh and Leichter can efficiently realize every access structure that can be described by a small monotone formula.[4]

Lemma 1. *Let $\mathcal{A}$ be an access structure and assume that $f_{\mathcal{A}}$ can be computed 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 $m \in \mathbb{N}$, $\mathcal{A}$ can be realized with domain of secrets $\mathbb{Z}_m$ by the scheme of [14]. The resulting scheme has information ratio $\max_{1 \leq j \leq n} a_j$.*

[4] A monotone formula is a formula with OR and AND gates without negations and without negated variables. The size of such formula is the number of leaves in the tree describing the formula. A monotone formula computes a monotone function.

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 [14]. However, for most 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.

3.5 The monotone Span Programs Construction [21, 46]

All the above constructions are linear, that is, the distribution scheme is a linear mapping. More formally, in a linear secret-sharing scheme over a finite field $\mathbb{F}$, the secret is an element of the field, the random string is a vector over the field such that each coordinate of this vector is chosen independently with uniform distribution from the field, and each share is a vector over the field such that each coordinate of this vector is some fixed linear combination of the secret and the coordinates of the random string.

Example 2. Consider the scheme for $\mathcal{A}_{\text{ustcon}}$ described in Section 3.2. This scheme is linear over the field with two elements $\mathbb{F}_2$. In particular, the randomness is a vector $\langle r_2, \ldots, r_{|V|-1} \rangle$ of $|V| - 2$ random elements in $\mathbb{F}_2$, and the share of an edge (v_1, v_2), for example, is $(k + r_2) \bmod 2$, that is, this is the linear combination where the coefficient of k and r_2 are 1 and all other coefficients are zero.

To model a linear scheme, we use *monotone span programs*, which is, basically, the matrix describing the linear mapping of the linear scheme. The monotone span program also defines the access structure which 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 $\mathbf{r}M$, then $\mathbf{r}$ is a row vector, if we write $M\mathbf{r}$, then $\mathbf{r}$ is a column vector).

Definition 4 (Monotone Span Program [46]). *A monotone span program is a triple $\mathcal{M} = (\mathbb{F}, M, \rho)$, where $\mathbb{F}$ is a field, M is an $a \times b$ matrix over $\mathbb{F}$, and $\rho : \{1, \ldots, a\} \to \{p_1, \ldots, p_n\}$ labels each row of M by a party.*[5] *The size of $\mathcal{M}$ is the number of rows of M (i.e., a). For any set $A \subseteq \{p_1, \ldots, p_n\}$, let M_A denote the sub-matrix obtained by restricting M to the rows labeled by parties in A. We say that $\mathcal{M}$ accepts B if the rows of M_B span the vector $\mathbf{e_1} = (1, 0, \ldots, 0)$. We say that $\mathcal{M}$ accepts an access structure $\mathcal{A}$ if $\mathcal{M}$ accepts a set B iff $B \in \mathcal{A}$.*

Example 3. Consider the following monotone span program $(\mathbb{F}_{17}, M, \rho)$, where

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}$$

[5] For simplicity, in this survey we label a row by a party p_j rather than by a variable x_j as done in [46].

and $\rho(1) = \rho(2) = p_2$, $\rho(3) = p_1$, and $\rho(4) = p_3$. Consider the sets $B = \{p_1, p_2\}$ and $T = \{p_1, p_3\}$. In this case

$$M_B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \text{ and } M_T = \begin{pmatrix} 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}.$$

As M_B has full rank, the rows of M_B span $\mathbf{e_1}$, i.e., $(3, 14, 1)M_B = \mathbf{e_1}$ (in $\mathbb{F}_{17}$). Hence, the span program accepts $\{p_1, p_2\}$. On the other hand, the rows of M_T do not span $\mathbf{e_1}$ and the span program does not accept $\{p_1, p_3\}$. The minimal authorized sets in the access structure accepted by $\mathcal{M}$ 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 2 ([21, 46]). *Let $\mathcal{M} = (\mathbb{F}, M, \rho)$ be a monotone span program accepting an access structure $\mathcal{A}$, where $\mathbb{F}$ is a finite field and for every $j \in \{1, \ldots, n\}$ there are a_j rows of M labeled by p_j. Then, there is a linear secret-sharing scheme realizing $\mathcal{A}$ such that the share of party p_j is a vector in $\mathbb{F}^{a_j}$. The information ratio of the resulting scheme is $\max_{1 \leq j \leq n} a_j$,*

Proof. Given the monotone span program $\mathcal{M} = (\mathbb{F}, M, \rho)$, where M is an $a \times b$ matrix over $\mathbb{F}$, define a linear secret-sharing scheme as follows:

- **Input:** a secret $k \in \mathbb{F}$.
- Choose $b - 1$ random elements $r_2, \ldots, r_b$ independently with uniform distribution from $\mathbb{F}$ and define $\mathbf{r} = (k, r_2, \ldots, r_b)$.
- Evaluate $(s_1, \ldots, s_a) = M\mathbf{r}$, and distribute to each player p_j the a_j entries corresponding to rows labeled by p_j.

In this linear secret-sharing scheme, every set in $\mathcal{A}$ can reconstruct the secret: Let $B \in \mathcal{A}$ and $N = M_B$, 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 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

$$\mathbf{v}(N\mathbf{r}) = (\mathbf{v}N)\mathbf{r} = \mathbf{e_1} \cdot \mathbf{r} = k.$$

We next prove that this scheme is private. If $T \notin \mathcal{A}$, then the rows of M_T do not span the vector $\mathbf{e_1}$, i.e., $\text{rank}(M_T) < \text{rank}\binom{M_T}{\mathbf{e_1}}$ (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, $|\text{kernel}(M_T)| > \left|\text{kernel}\binom{M_T}{\mathbf{e_1}}\right|$, and there is some vector $\mathbf{w} \in \mathbb{F}^b$ such that $(M_T)\mathbf{w} = \mathbf{0}$ and $\mathbf{e_1} \cdot \mathbf{w} = 1$ (that is, $w_1 = 1$). We next prove that for every vector of shares $(s_1, \ldots, s_{|T|})$ for the parties in T, the probability that it is generated is the same for every secret $k \in \mathbb{F}$. Fix a vector $\mathbf{r} = (0, r_2, \ldots, r_b)$ such that $(M_T)\mathbf{r} = (s_1, \ldots, s_{|T|})$, that is, $\mathbf{r}$ is a vector generating shares for the secret $k = 0$. For any $k \in \mathbb{F}$ and consider the vector $\mathbf{r'} = \mathbf{r} + k\mathbf{w}$. As $r'_1 = k$, the vector $\mathbf{r'}$ generates shares for the secret k. Furthermore,

$$(M_T)\mathbf{r'} = (M_T)(\mathbf{r} + k\mathbf{w}) = (M_T)\mathbf{r} + k(M_T)\mathbf{w} = (M_T)\mathbf{r} = (s_1, \ldots, s_{|T|}).$$

That is, for every $k \in K$ the number of random strings that generate the shares $(s_1, \ldots, s_{|T|})$ when the secret is k is the same as the number of random strings that generate these shares when the secret is 0, and the scheme is private. □

Remark 2 (Historical Notes). Brickell [21] in 1989 implicitly defined monotone span programs for the case that each party labels exactly one row, and proved Claim 2. Karchmer and Wigderson [46] 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 [3] proved that linear secret-sharing schemes 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. We next describe the linear secret-sharing for $\mathcal{A}_{\text{ustcon}}$, presented in Section 3.2, as a monotone span program. In this access structure, 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 every party (edge) (v_i, v_j), where $1 \leq i < j \leq m-1$, there is a unique row in the program labeled by this party; in this row all entries in the row are zero, except for the ith and the jth entries which are 1. Furthermore, for every 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; in this row all entries in the row are zero, except for the ith entry which is 1 (this is equivalent to choosing $r_m = 0$ in Section 3.2). 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 $\mathbf{r} = (k, r_2, \ldots, r_{m-1})$ 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 $\mathcal{A}_{\text{ustcon}}$ described above.

3.6 Multi-Linear Secret-Sharing Schemes [16, 32]

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 [16, 32], are called multi linear and are based on the following generalization of monotone span programs.

Definition 5 (Multi-Target Monotone Span Program). *A multi-target monotone span program is a quadruple $\mathcal{M} = (\mathbb{F}, M, \rho, V)$, where $\mathbb{F}$ is a finite field, M is an $a \times b$ matrix over $\mathbb{F}$, $\rho : \{1, \ldots, a\} \to \{p_1, \ldots, p_n\}$ labels each row of M by a party, and $V = \{\mathbf{e_1}, \ldots, \mathbf{e_c}\}$ is a set of vectors in $\mathbb{F}^b$ for some $1 \leq c < b$ such that for every $A \subseteq \{p_1, \ldots, p_n\}$ either*

- *The rows of M_A span each vector in $\{\mathbf{e_1}, \ldots, \mathbf{e_c}\}$. In this case, we say that $\mathcal{M}$ accepts A, or,*

– *The rows of M_A span no non-zero vector in the linear space spanned by* $\{\mathbf{e_1}, \ldots, \mathbf{e_c}\}$.

We say that $\mathcal{M}$ accepts an access structure $\mathcal{A}$ if $\mathcal{M}$ accepts a set B iff $B \in \mathcal{A}$.

Claim 3. *Let $\mathcal{M} = (\mathbb{F}, M, \rho, V)$ be a multi-target monotone span program accepting $\mathcal{A}$, where $\mathbb{F}$ 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 multi-linear secret-sharing scheme realizing $\mathcal{A}$ such that the secret is a vector in $\mathbb{F}^c$ and the share of party p_j is a vector in $\mathbb{F}^{a_j}$; in particular, the information ratio of the scheme is* $\max_{1 \leq j \leq n} a_j/c$.

The proof of Claim 3 is similar to the proof of Claim 2, where in this case the secret is $k_1, \ldots, k_c$, the dealer chooses $b - c$ random elements $r_{c+1}, \ldots, r_b$ in $\mathbb{F}$, uses the vector $\mathbf{r} = (k_1, \ldots, k_c, r_{c+1}, \ldots, r_b)$, and computes the shares $M\mathbf{r}$. Any multi-target monotone span program is a monotone span program; however, using it to construct a multi-linear secret-sharing scheme results in a scheme with better information ratio.

Example 5. Let $\sqcap$ be the access structure with 4 parties p_1, p_2, p_3, p_4 whose minimal authorized sets are $\{p_1, p_2\}, \{p_2, p_3\}, \{p_3, p_4\}$. It was proved by [23] that in any secret-sharing scheme realizing $\sqcap$ the information ratio is at least 1.5. We present this lower bound and prove it in Theorem 1. By definition, the information ratio of a linear scheme is integral. We next present a multi-linear secret-sharing scheme realizing $\sqcap$ with information ratio 1.5. We first describe a linear scheme whose information ratio is 2. To share a bit $k_1 \in \mathbb{F}_2$, the dealer independently chooses two random bits r_1 and r_2 with uniform distribution. The share of p_1 is r_1, the share of p_2 is $r_1 \oplus k_1$, the share of p_3 is two bits, r_1 and $r_2 \oplus k_1$, and the share of p_4 is r_2. Clearly, this scheme realizes $\sqcap$.

Notice that although p_2 and p_3 have symmetric roles in $\sqcap$, in the above scheme p_2 gets one bit and p_3 gets two bits. To construct a multi-linear scheme realizing $\sqcap$ whose information ratio is 1.5, we exploit the asymmetry of the previous scheme. To share a secret $(k_1, k_2) \in (\mathbb{F}_2)^2$, the dealer independently chooses four random bits r_1, r_2, r_3, and r_4 with uniform distribution. The scheme is described in the following table.

Share of p_1	Share of p_2	Share of p_3	Share of p_4
r_1, r_3	$r_1 \oplus k_1, r_3 \oplus k_2, r_4$	$r_1, r_2 \oplus k_1, r_4 \oplus k_2$	r_2, r_4

The secret in the above scheme is two bits and the largest shares are 3 bits, hence the information ratio of this scheme is 1.5. It is an easy exercise to write the above multi-linear scheme as a multi-target monotone span program; the matrix of this program has 10 rows and 6 columns.

The scheme in Example 5 involves two applications of linear secret-sharing schemes realizing $\sqcap$, each application with an independent secret. In particular, the multi-linear secret-sharing scheme has the same average information ratio as the linear scheme. Simonis and Ashikhmin [62] construct a multi-linear secret-sharing scheme realizing some access structure with information ratio and average information ratio 1. Furthermore, using results on representation of matroids,

they prove that any linear secret-sharing scheme realizing this access structure has average information ratio greater than 1. Thus, multi-linear secret-sharing schemes are more efficient than linear secret-sharing schemes. The maximum possible improvement in the information ratio and average information ratio of multi-linear secret-sharing schemes compared to linear secret-sharing schemes is open.

3.7 Other Constructions

There are many other constructions of secret-sharing schemes for other specific access structures, e.g., hierarchical access structures [60, 21, 64, 66], weighted threshold access structures [11], and more complicated compositions of access structures [63, 34].

4 Secret Sharing and Secure Multi-Party Computation

Secret-sharing schemes are a basic building box in construction of many cryptographic protocols. In this section we demonstrate the use of secret-sharing schemes for secure multi-party computation of general functions. For simplicity we concentrate on the case that the parties are honest-but-curious, 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 supply information-theoretic security.

Definition 6 (Secure Computation in the Honest-but-Curious Model (Informal)). *Let $\mathbb{F}$ be a finite field. Assume there are n parties $p_1, \ldots, p_n$, and at most t of them are corrupted, where $t < n$. Each party p_j holds a private input $x_j \in \mathbb{F}$. The parties want to compute some function $f(x_1, \ldots, x_n)$ by exchanging messages on private channels according to some protocol $\mathcal{P}$. We have two requirements:*

Correctness. *At the end of the protocol each party outputs $f(x_1, \ldots, x_n)$.*
Privacy. *Every coalition T of at most t parties cannot learn any information not implied by the inputs $\{x_j\}_{p_j \in T}$ and the output of the function.*

We will first show a homomorphic property of Shamir's secret-sharing scheme. Using this property, we show how to use secret sharing to construct a protocol for securely computing the sum of secret inputs. Then, we will show how to securely compute the product of inputs. Combining these protocols we get an efficient protocol for computing any function which can be computed by a small arithmetic circuit. Such protocols with information-theoretic security were first presented in [13, 24]. The exact protocol we present here is from [38].

Claim 4. *Let $k_1, k_2 \in \mathbb{F}$ be two secrets. For $i \in \{1, 2\}$, let $s_{i,1}, \ldots, s_{i,n}$ be a sharing of k_i using Shamir's $(t+1)$-out-of-n scheme (see Section 3.1). Then,*

$s_{1,1}+s_{2,1},\ldots,s_{1,n}+s_{2,n}$ are shares of the secret k_1+k_2 in Shamir's $(t+1)$-out-of-n scheme. Similarly, $s_{1,1}\cdot s_{2,1},\ldots,s_{1,n}\cdot s_{2,n}$ are shares of the secret $k_1\cdot k_2$ 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 $s_{1,1},\ldots,s_{1,n}$ and $s_{2,1},\ldots,s_{2,n}$ respectively, that is $Q_i(0)=k_i$ and $Q_i(\alpha_j)=s_{i,j}$ for $i\in\{1,2\}$ and $1\leq j\leq n$ (where $\alpha_1,\ldots,\alpha_n$ are defined in Section 3.1). Define $Q(x)=Q_1(x)+Q_2(x)$. This is a polynomial of degree at most t such that $Q(0)=Q_1(0)+Q_2(0)=k_1+k_2$ and $Q(\alpha_j)=s_{1,j}+s_{2,j}$, that is, this is a polynomial generating the shares $s_{1,1}+s_{2,1},\ldots,s_{1,n}+s_{2,n}$ given the secret k_1+k_2.

Similarly, let $R(x)=Q_1(x)\cdot Q_2(x)$. This is a polynomial degree at most $2t$ generating the shares $s_{1,1}\cdot s_{2,1},\ldots,s_{1,n}\cdot s_{2,n}$ given the secret $k_1\cdot k_2$.[6] □

4.1 Computing the Sum of Two Shared Numbers

Assume that two secrets x_1 and x_2 are shared using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Using Claim 4, each party can compute a share of the sum of the secrets without any communication.

Input of party p_j. Shares $s_{1,j}$ and $s_{2,j}$ of the secrets x_1 and x_2 respectively.
Computation step: Each party p_j computes $s_j=s_{1,j}+s_{2,j}$.

4.2 Computing the Product of Two Shared Numbers

Assume that two secrets x_1 and x_2 are shared using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Using Claim 4, the parties can compute shares of the product $x_1\cdot x_2$ in a $(2t+1)$-out-of-n secret-sharing scheme. We show that, by using 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 corrupt parties, where $n=2t+1$ (that is, there is a majority of honest parties).

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

Step I. Each party p_j computes $s_j=s_{1,j}\cdot s_{2,j}$ and shares s_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 $\beta_1,\ldots,\beta_\ell$ be the constants defined in (7) for the reconstruction of the secret in Shamir's $(2t+1)$-out-of-n scheme. Each party p_ℓ computes $u_\ell=\sum_{j=1}^{n}\beta_j q_{j,\ell}$.

We next explain why this protocol is correct. By Claim 4, $s_1,\ldots,s_n$ are shares of $x_1\cdot x_2$ in a Shamir's $(2t+1)$-out-of-n scheme. Thus, by (7), $x_1\cdot x_2=\sum_{j=1}^{n}\beta_j s_j$. As $q_{j,1},\ldots,q_{j,n}$ are shares in Shamir's $(t+1)$-out-of-n scheme of the secret s_j, Claim 4 implies that $u_1,\ldots,u_\ell$ are shares of $x_1\cdot x_2$.

[6] While $Q(x)$ is a uniformly distributed polynomial such that $Q(0)=k_1+k_2$, 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.

4.3 Computing an Arithmetic Circuit

Using the above protocols, we show how to securely compute any function represented by an arithmetic circuit assuming that $n = 2t+1$. 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, called input gates. For each i, where $1 \leq i \leq n$, there is a node labeled by the variable x_i.[7]
- Each internal node is labeled either by $\times$, called a multiplication gate, or by $+$, called an addition gate. Each internal node has in-degree two.

The function computed by an arithmetic circuit over a field $\mathbb{F}$ is defined in the natural way, where the arithmetic is done over $\mathbb{F}$. The complexity of computing the function is proportional to the number of gates in the circuit. We next show a secure protocol for evaluating the function computed by an arithmetic circuit, where each party p_j holds x_j. The number of rounds in this protocol is linear in the number of gates. More formally, let $G_1, G_2, \ldots, G_\ell$ be the gates of a circuit sorted according to some topological order (that is, if there exists an edge from G_j to G_i, then $i > j$). Assume that, for $1 \leq i \leq n$, the gate G_i is labeled by x_i.

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 phases, where in the beginning of phase i the parties hold shares of a $(t+1)$-out-of-n secret-sharing scheme of the two inputs of gate G_i, and in the end of phase i the parties hold shares of a $(t+1)$-out-of-n secret-sharing scheme of the output of the gate G_i. At the end of the protocol, the output is reconstructed from the shares.

Input of party p_j. An element $x_j \in \mathbb{F}$.

Initialization. Each party p_i shares x_i using Shamir's $(t+1)$-out-of-n secret-sharing scheme. Denote the resulting shares by $q_{i,1}, \ldots, q_{i,n}$. Party p_i sends $q_{i,j}$ to p_j.

Computation stages. For $i = n+1$ to ℓ compute shares of the output of gate G_i as follows:

- Assume that the incoming edges into gate G_i are from gates G_j and G_k, where $j, k < i$ and the parties holds shares $q_{j,1}, \ldots, q_{j,n}$ and $q_{k,1}, \ldots, q_{k,n}$ of the outputs of these gates.
- If G_i is an addition gate, each party p_m locally computes $q_{i,m} = q_{j,m} + q_{k,m}$ as the share of the output of gate G_i.
- If G_i is a multiplication gate, the parties use the one-round protocol described in Section 4.2 to compute shares of the product of the outputs of gates G_j and G_k.

[7] There can be additional nodes with in-degree 0 labeled by constants. For simplicity, we ignore such nodes.

Reconstruction. Each party p_m sends its share $q_{\ell,m}$ to p_1. Party p_1 reconstructs a secret s from the shares $q_{\ell,1}, \ldots, q_{\ell,t+1}$ using the reconstruction procedure of Shamir's $(t+1)$-out-of-n secret-sharing scheme, and sends s to all parties, which output this value.

By the correctness of the addition and multiplication protocols, at the end of phase i, the parties hold shares of the output of gate 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, 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, thus, the set does not learn information not implied by the inputs of the set and the sum.

4.4 Extensions to Other Models

The protocol we described above assumes that the corrupted parties are honest-but-curious. A more realistic assumption is that the parties can deviate from the protocol and send any messages that might help them. Such parties are called malicious. For example, in the multiplication protocol, a party that should share s_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. [25]. Such schemes were constructed under various assumptions, see [38] for a partial list of such constructions. We will not elaborate on verifiable secret sharing in this survey.

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 [42] 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 require 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 honest-but-curious 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 [42]. For every such access structure $\mathcal{A}$, Cramer et al. [28] showed that using every linear secret-sharing scheme realizing $\mathcal{A}$, 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 is also similar, however, the choice of the constants $\beta_1, \ldots, \beta_n$ is more involved. The protocol of Cramer et al. [28] shows the need for general secret-sharing schemes.

5 Lower Bounds on the Size of the Shares

The best known constructions of secret-sharing schemes for general access structures (e.g., [45, 14, 21, 46, 16, 32]) have information ratio $2^{O(n)}$, where n is the number of parties in the access structure. As discussed in the introduction, we conjecture that this is the best possible. Lower bounds for secret-sharing schemes have been proved in, e.g., [47, 23, 19, 33, 29, 30, 18]. However, these lower bounds are far from the exponential upper bounds. The best lower bound was proved by Csirmaz [29, 30], who proved that for every n there exists an n-party access structure such that every secret-sharing scheme realizing it has information ratio $\Omega(n/\log n)$. In Sections 5.2 – 5.3, we review this proof. For linear secret-sharing schemes the situation is much better – for every n there exist access structures with n parties such that every linear secret-sharing scheme realizing them has super-polynomial, i.e., $n^{\Omega(\log n)}$, information ratio [5, 2, 36, 37]. In Section 5.5, we present the lower bound proof of [37].

5.1 A Simple Lower Bound

Karnin et al. [47] have showed that for each non-redundant party p_j (that is, a party that appears in at least one minimal authorized set) $H(S_j) \geq H(S)$, which 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.

Lemma 2. *Let p_j be a non-redundant party in $\mathcal{A}$ and let Σ be any secret-sharing scheme realizing $\mathcal{A}$, where K and K_j are the domains of secrets and of the shares of p_j respectively. Then, $|K_j| \geq |K|$.*

Proof. Let B be a minimal authorized set in $\mathcal{A}$ containing p_j, that is $B \in \mathcal{A}$ and $B' \stackrel{\text{def}}{=} B \setminus \{p_j\} \notin \mathcal{A}$. Assume that there is a secret-sharing-scheme realizing $\mathcal{A}$ in which $|K_j| < |K|$. Fix any vector of shares $\{s_i\}_{p_i \in B'}$ for the parties of B' that has positive probability (given some secret $k_0 \in K$). By the privacy property, this vector of shares should have positive probability given any secret $k \in K$. That is, for every $k \in K$, there is a share $s^k \in K_j$ such that $\{s_i\}_{p_i \in B'}$ together with s^k have positive probability given the secret k. Since $|K_j| < |K|$, there are secrets $k_1, k_2 \in K$ such that $k_1 \neq k_2$ and $s^{k_1} = s^{k_2}$. Thus, the authorized set B holding the shares $\{s_i\}_{p_i \in B'}$ and s^{k_1} errs in the reconstruction for at least one of the secrets k_1 and k_2, contradicting the correctness of the scheme. □

5.2 Stronger Lower Bounds

Starting from the works of Karnin et al. [47] and Capocelli et al. [23], the entropy was used to prove lower bounds on the share size in secret-sharing schemes [19, 33, 29, 30]. In other words, to prove lower bounds on the information ratio of secret-sharing schemes, we use the alternative definition of secret sharing via the entropy function, Definition 3.

Towards proving lower bounds, we use properties of the entropy function as well as the correctness and privacy of secret-sharing schemes. This is summarized in Claim 5. To simplify notations, in the sequel we denote $H(S_A)$ by $H(A)$ for any set of parties $A \subseteq \{p_1, \ldots, p_n\}$. Furthermore, we denote $H(S_A S)$ by $H(AS)$. In the lower bounds proof, we assume uniform distribution on the secrets, that is, $H(S) = \log |K|$. As proved in Claim 1, this assumption is without loss of generality. By the properties of the entropy function, for every j, $H(\{p_j\}) \leq \log |K_j|$, thus, the information ratio of the scheme, that is, $\max_{1 \leq j \leq n} \log |K_j| / \log |K|$, is at least $\max_{1 \leq j \leq n} H(\{p_j\})/H(S)$.

Claim 5. *Let $A, B \subseteq \{p_1, \ldots, p_n\}$ and Σ be a secret-sharing scheme realizing an access structure $\mathcal{A}$. 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 $A \notin \mathcal{A}$, $B \in \mathcal{A}$, and $A \subset B$, then* $H(B) \geq H(A) + H(S)$.
Strong Submodularity.	*If $A, B \in \mathcal{A}$ and $A \cap B \notin \mathcal{A}$, 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 (where the submodularity follows from the fact that the conditional mutual information is non-negative). For the strong monotonicity observe that by the correctness, monotonicity, and privacy, $H(B) = H(BS) \geq H(AS) = H(A) + H(S)$. For the strong submodularity, note that if $A, B \in \mathcal{A}$ and $A \cap B \notin \mathcal{A}$, 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, $H(A) + H(B) = H(AS) + H(BS) \geq H((A \cup B)S) + H((A \cap B)S) = H(A \cup B) + H(A \cap B) + H(S)$. □

To give an example of using Claim 5, we present the lower bound of [23] for the access structure $\sqcap$ (defined in Example 5).

Theorem 1 ([23]). *The information ratio of every secret-sharing scheme realizing $\sqcap$ is at least* 1.5.

Proof. Let Σ be any secret-sharing scheme realizing $\sqcap$. By Claim 5,

$$H(\{p_1, p_2\}) + H(\{p_2, p_3\}) \geq H(\{p_1, p_2, p_3\}) + H(\{p_2\}) + H(S), \tag{8}$$

$$H(\{p_1, p_3, p_4\}) \geq H(\{p_1, p_4\}) + H(S), \tag{9}$$

$$H(\{p_1, p_2, p_3\}) \geq H(\{p_1, p_3\}) + H(S), \tag{10}$$

$$H(\{p_1, p_3\}) + H(\{p_1, p_4\}) \geq H(\{p_1, p_3, p_4\}) + H(\{p_1\}), \tag{11}$$

$$H(\{p_1\}) + H(\{p_2\}) \geq H(\{p_1, p_2\}), \tag{12}$$

$$H(\{p_2\}) + H(\{p_3\}) \geq H(\{p_2, p_3\}). \tag{13}$$

In the above, (8) follows from strong submodularity, (9) and (10) follow from strong monotonicity, and (11), (12), and (13) follow from submodularity. 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

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

□

5.3 Csirmaz's Lower Bound

We next present Csirmaz's lower bound on the information ratio. We first define, for every $n \in \mathbb{N}$, an access structure $\mathcal{A}_n$ by specifying its minimal sets. Let k be the largest integer such that $2^k + k - 1 \leq n$. Let $B = \{p_1, \ldots, p_{2^k-1}\}$ and define $B_0 = \emptyset$ and $B_i = \{p_1, \ldots, p_i\}$ for $1 \leq i \leq 2^k - 1$. Furthermore, let $A = \{p_{2^k}, \ldots, p_{2^k+k-1}\}$ (that is, $|A| = k$), and $A = A_0, A_1, \ldots, A_{2^k-1} = \emptyset$ be all the subsets of A such that if $i < i'$, then $A_i \not\subseteq A_{i'}$. Finally, define $U_i = A_i \cup B_i$ for $0 \leq i \leq 2^k - 1$. The minimal sets of $\mathcal{A}_n$ are $U_0, U_1, \ldots, U_{2^k-1}$.

Lemma 3. *For every* $0 \leq i \leq 2^k - 2$

$$H(B_i \cup A) - H(B_i) \geq H(B_{i+1} \cup A) - H(B_{i+1}) + H(S). \tag{14}$$

Proof. On the one hand, $U_i \subseteq B_i \cup A \in \mathcal{A}_n$, and $U_{i+1} = B_{i+1} \cup A_{i+1} \in \mathcal{A}_n$. On the other hand, $B_i \cup A_{i+1} \notin \mathcal{A}_n$, since $B_i \cup A_{i+1}$ does not contain any minimal authorized set U_j:

Case I: $j > i$**.** $p_{i+1} \in U_j = B_j \cup A_j$, while $p_{i+1} \notin B_i \cup A_{i+1}$,
Case I: $j \leq i$**.** $A_j \subseteq U_j$, while $A_j \not\subseteq A_{i+1}$.

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 results in (14). □

Theorem 2 ([29]). *For every* n *there exists an* n*-party access structure* $\mathcal{A}_n$ *such that every secret-sharing scheme realizing it has information ratio* $\Omega(n/\log n)$*.*

Proof. Summing (14) for every $0 \leq i \leq 2^k - 2$ we get that

$$H(B_0 \cup A) - H(B_0) \geq H(B_{2^k-1} \cup A) - H(B_{2^k-1}) + (2^k - 1)H(S). \tag{15}$$

By monotonicity, $H(B_{2^k-1} \cup A) - H(B_{2^k-1}) \geq 0$. Furthermore, by submodularity, $\sum_{p_j \in A} H(\{p_j\}) \geq H(A)$ and $H(B_0) + H(A) \geq H(B_0 \cup A)$. Thus,

$$\begin{aligned}
\sum_{p_j \in A} H(\{p_j\}) &\geq H(A) \\
&\geq H(B_0 \cup A) - H(B_0) \\
&\geq H(B_{2^k-1} \cup A) - H(B_{2^k-1}) + (2^k - 1)H(S) \\
&= \Omega(n) \cdot H(S).
\end{aligned}$$

This implies that $H(\{p_j\}) = \Omega(n/\log n)H(S)$ for at least one party p_j, and the information ratio of the scheme is $\Omega(n/\log n)$. □

We next show how to strengthen Theorem 2 and show that there exists an access structure in which the shares of many parties have to be long.

Theorem 3 ([30]). *For every n there exists an n-party access structure $\mathcal{A}'_n$ such that every secret-sharing scheme realizing it has* average *information ratio $\Omega(n/\log n)$.*

Proof. In the proof of Theorem 2 we constructed an access structure in which 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 $\Omega(n)H(S)$. We next construct a similar access structure which has many copies of A and one copy of B. Let k be the largest integer such that $2^k \leq n/2$. Let $B = \{p_1, \ldots, p_{2^k-1}\}$ and define $B_0 = \emptyset$ and $B_i = \{p_1, \ldots, p_i\}$ for $1 \leq i \leq 2^k - 1$. Furthermore, let $A^\ell = \left\{p_{2^k+\ell k}, \ldots, p_{2^k+(\ell+1)k-1}\right\}$ for $0 \leq \ell \leq \lfloor n/2k \rfloor - 1$, and $A^\ell = A^\ell_0, A^\ell_1, \ldots, A^\ell_{2^k-1} = \emptyset$ be all the subsets of A^ℓ such that if $i < i'$, then $A^\ell_i \not\subseteq A^\ell_{i'}$. Finally, the minimal sets of $\mathcal{A}'_n$ are $U^\ell_i \stackrel{\text{def}}{=} A^\ell_i \cup B_i$ for $0 \leq i \leq 2^k - 1$ and $0 \leq \ell \leq \lfloor n/2k \rfloor - 1$.

For every ℓ, the access structure $\mathcal{A}'_n$ restricted to the parties in $B \cup A^\ell$ is isomorphic to the access structure $\mathcal{A}_{n'}$ (where $n' > n/2$). Thus, by (16),

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

As the sets A^ℓ are disjoint,

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

Thus, the average information ratio of $\mathcal{A}'_n$ is $\Omega(n/\log n)$. □

5.4 Limitations of Known Techniques for Lower Bounds

Basically, all known lower bounds for the size of shares in secret-sharing schemes are implied by Claim 5. In other words, they only use the so-called Shannon information inequalities (i.e., the fact that the conditional mutual information is non-negative). Csirmaz [29] in 1994 proved that such proofs cannot prove a lower bound of $\omega(n)$ on the information ratio. That is, Csirmaz's lower bound is nearly optimal (up to a factor $\log n$) using only Shannon inequalities. In 1998, new information inequalities were discovered by Zhang and Yeung [71]. Other information inequalities were discovered since, see, e.g. [70]. In particular, there are infinitely many independent information inequalities in 4 variables [50]. Such inequalities were used in [7, 51] to prove lower bounds for secret-sharing schemes. Beimel and Orlov [8] proved that all information information inequalities with 4 or 5 variables and all known information inequalities in more than 5 variables cannot prove a lower bound of $\omega(n)$ on the information ratio of secret-sharing schemes. Thus, new information inequalities with more than 5 variables should be found if we want to improve the lower bounds.

5.5 Lower Bounds for Linear Secret Sharing

For linear secret-sharing schemes we can prove much stronger lower bounds than for general secret-sharing schemes. Recall that linear secret-sharing schemes are equivalent to monotone span programs and we first state the results using monotone span programs. Lower bounds for monotone span programs were presented in [5, 2, 36, 37, 10]; the best known lower bound is $n^{\Omega(\log n)}$ as proved in [36]. We present here an alternative proof of [37]. We start with a simple observation.

Observation 1. Let $\mathcal{A}$ be a (monotone) access structure. Let $B \in \mathcal{A}$ and $C \subseteq \{p_1, \ldots, p_n\}$ such that $\{p_1, \ldots, p_n\} \setminus C \notin \mathcal{A}$. 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 $B \in \mathcal{A}$ and $\{p_1, \ldots, p_n\} \setminus C \notin \mathcal{A}$.

To prove the lower bound, Gál and Pudlák [37] choose 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 $\mathcal{A}$.

Let $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ be the collection of minimal authorized sets in $\mathcal{A}$ and $\mathcal{C} = \{(C_{1,0}, C_{1,1}), (C_{2,0}, C_{2,1}), \ldots, (C_{t,0}, C_{t,1})\}$ be a collection of pairs of sets of parties such that $\{p_1, \ldots, p_n\} \setminus (C_{j,0} \cup C_{j,1}) \notin \mathcal{A}$ for every $1 \leq j \leq t$. By Observation 1, $B_i \cap (C_{j,0} \cup C_{j,1}) \neq \emptyset$ 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. *We say that a collection $\mathcal{C}$ satisfies the* unique intersection *property for $\mathcal{A}$ if*

1. *For every $1 \leq j \leq t$, $\{p_1, \ldots, p_n\} \setminus (C_{j,0} \cup C_{j,1}) \notin \mathcal{A}$.*
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 6. 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 $\mathcal{C}$ satisfying the unique intersection property for $\mathcal{A}$, where $\mathcal{C}$ is $(\{p_1, p_2\}, \{p_{10}\})$, $(\{p_1, p_3\}, \{p_9\})$, $(\{p_1, p_4\}, \{p_8\})$, $(\{p_2, p_3\}, \{p_7\})$, $(\{p_2, p_4\}, \{p_6\})$, and $(\{p_3, p_5\}, \{p_1\})$.

It can be seen that $\mathcal{C}$ satisfies (1). 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 since the only minimal authorized set that contains $\{p_3, p_4\}$ is $\{p_3, p_4, p_{10}\}$ and $p_{10} \notin T$. Furthermore, $\mathcal{C}$ satisfies (2). Consider, e.g., $\{p_1, p_3, p_6\} \in \mathcal{B}$ and $(\{p_1, p_2\}, \{p_{10}\}) \in \mathcal{C}$. 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 4 ([37]). *Let $\mathcal{C}$ be a collection satisfying the unique intersection property for $\mathcal{A}$ 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 $\mathcal{A}$ is at least* $\mathrm{rank}_{\mathbb{F}}(D)$.

Example 7. The matrix D defined for the set $\mathcal{C}$ of Example 6 is the full rank matrix described below:

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

Proof (Proof of Theorem 4). Let $\mathcal{M} = (\mathbb{F}, M, \rho)$ be a monotone span program accepting $\mathcal{A}$, and denote the size of $\mathcal{M}$ (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, $\text{rank}_{\mathbb{F}}(D) \leq \text{rank}_{\mathbb{F}}(L) \leq m$.

Fix any i such that $1 \leq i \leq \ell$. Since $B_i \in \mathcal{A}$, 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 $T_j \notin \mathcal{A}$, the rows in M labeled by the parties in T_j do not span the vector $\mathbf{e_1}$. As explained in Section 3.4, 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, only coordinates in $\mathbf{z_j}$ labeled by parties in $C_{j,1}$ can be non-zero. Hence,

- If $B_i \cap C_{j,0} \neq \emptyset$, then $D_{i,j} = 0$ 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, 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, $\text{rank}_{\mathbb{F}}(D) = \text{rank}_{\mathbb{F}}(LR) \leq \text{rank}_{\mathbb{F}}(L) \leq m$. In other words, the rank of D is at most the size of smallest monotone span program accepting $\mathcal{A}$. □

We next present a construction of an access structure for which we can prove an $n^{\Omega(\log n)}$ lower bound using Theorem 4. A 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 a 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 8. Consider the graph described in Figure 1. This is a trivial graph satisfying the isolated neighbor property for $t = 2$. 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 6.

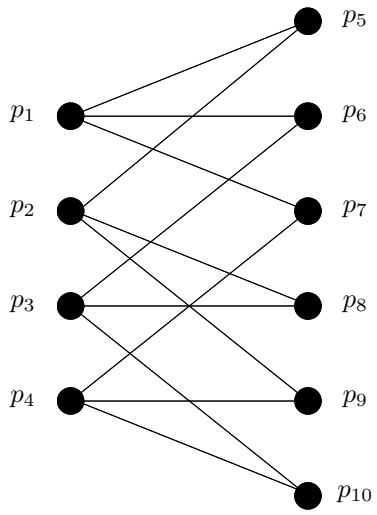

Fig. 1. An example of a graph satisfying the isolated neighbor property for $t = 2$.

Lemma 4. *If G has the isolated neighbor property for t, then the size of every monotone span program accepting $\mathcal{N}_G$ is at least $\binom{|U|}{t}$.*

Proof. We prove the lemma using Theorem 4. We take $\mathcal{C}$ to be all the pairs C_0, C_1, where $C_0 \subset U$ such that $|C_0| = t$ and $C_1 = \{v : \forall_{u \in C_0}(u, v) \notin E\}$, 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 $\mathcal{A}$:

- 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 $T \notin \mathcal{A}$, that is, T does not contain any minimal authorized set. Let $A \subseteq U \cap T$ be any set such that $|A| = t$. Thus, $|A| = |C_0| = t$, and there is a vertex

$v \in V$ such that $v \in N(A)$ and $v \in C_1$, 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 $(C_0, C_1) \in \mathcal{C}$. 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 $v \in C_1$ is not a neighbor of u, and $N(A) \cap C_0 = \emptyset$.

Thus, by Theorem 4, the size of every monotone span program accepting $\mathcal{A}$ 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 (C_0, C_1) 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 (see, e.g., [48, Example 2.12]).[8] The rank of D is, thus, the number of minimal authorized sets in $\mathcal{A}$, namely, $\binom{|U|}{t}$. □

As there exist graphs which satisfy the isolated neighbor property for $t = \Omega(\log n)$, e.g., the Paley Graph [1], we derive the promised lower bound.

Theorem 5. *For every n, there exists an access structure $\mathcal{N}_n$ such that every monotone span program over any field accepting it has size $n^{\Omega(\log n)}$.*

As monotone span program are equivalent to linear secret-sharing schemes [46, 3], the same lower bound applies to linear secret-sharing schemes.

Corollary 1. *For every n, there exists an access structure $\mathcal{N}_n$ such that the information ratio of every linear secret-sharing scheme realizing it is $n^{\Omega(\log n)}$.*

In multi-linear secret-sharing schemes, the secret can be a vector of elements over $\mathbb{F}$, which can reduce the information ratio. However, every multi-linear secret-sharing scheme implies a linear scheme with the same share length. Thus, we obtain the following lower bound.

Corollary 2. *For every n, there exists an access structure $\mathcal{N}_n$ such that the length of the shares in every multi-linear secret-sharing scheme realizing it over the field $\mathbb{F}$ is $n^{\Omega(\log n)} \log |\mathbb{F}|$.*

Thus, already for moderate values of n, we get that the size of shares in any multi-linear secret-sharing scheme realizing $\mathcal{N}_n$ is impractical.

6 Secret-Sharing, Cryptomania, and $NP \neq coNP$

In this section we describe two unpublished results of Rudich [56], showing two surprising results connecting secret-sharing schemes to two fundamental questions in cryptography and complexity.

[8] The proof in [48] requires that $n - 2t + 1 \neq 0$ in the field $\mathbb{F}$.

6.1 Impossibility of Secret-Sharing with Efficient Sharing

In previous sections we said that a scheme is efficient if the length of the shares is polynomial in the number of parties in the access structures. This is only a necessary condition for being efficient. To use secret-sharing schemes, we should also require that the sharing process and the reconstruction are efficient. That is, when using secret-sharing schemes we want the honest parties, which share secrets and reconstruct them, to run in polynomial time.

We first consider secret-sharing where the dealer is efficient. Formally, a secret-sharing scheme $\langle \Pi, \mu \rangle$ has efficient sharing if there is an algorithm computing the mapping Π whose running time is polynomial in n (the number of parties in the access structure) and $\log |K|$ (the number of bits in the secret). Rudich [56] proved that it is unlikely that there is a secret-sharing scheme with efficient sharing realizing the Hamiltonian access structure, $\mathcal{A}_{\text{ham}}$, defined below, that is, assuming that $NP \neq coNP$, there is no such a scheme.

Definition 8. *A Hamiltonian cycle in an undirected graph $G = (V, E)$ is a simple cycle passing through all vertices in V. The Hamiltonian access structure, denoted $\mathcal{A}_{\text{ham}}$, is the access structure whose parties are edges of a complete undirected graph and its authorized sets are subsets of the edges containing a Hamiltonian cycle.*

Theorem 6 ([56]). *If $NP \neq coNP$, then there is no secret-sharing scheme with efficient reconstruction realizing $\mathcal{A}_{\text{ham}}$.*

Proof. Let $\overline{\text{HAM}} \stackrel{\text{def}}{=} \{G : G \text{ does not contain a Hamiltonian cycle}\}$. We assume in a way of contradiction that there is a secret-sharing scheme with efficient reconstruction realizing $\mathcal{A}_{\text{ham}}$ and prove that $NP = coNP$, that is, we prove that $\overline{\text{HAM}} \in NP$. The proof relies on the following simple observation: A graph $G = (V, E)$ does not contain a Hamiltonian cycle, that is, $E \notin \mathcal{A}_{\text{ham}}$, iff the shares of the parties in E do not determine the secret iff the shares of the parties in E could be generated both for the secret 0 and for the secret 1. Now, given a graph G, the witness that $G \in \overline{\text{HAM}}$ is two random strings r_0 and r_1 such that the scheme with secret $k = 0$ and random string r_0 produces the same shares for the parties in E as the scheme with secret $k = 1$ and random string r_1. □

In the above theorem, we require a very weak privacy requirement, that is, for every unauthorized set there exists shares that could be generated both for the secret 0 and the secret 1. Furthermore, we only require that the sharing is efficient and we do not care if the reconstruction is efficient. However, we require perfect correctness, that is, an authorized set can always reconstruct the correct secret.

6.2 Oblivious-Transfer Protocols from Secret-Sharing

To appreciate the result presented below we start with some background. Cryptographic protocols are built based on some assumptions. These assumption can

be specific (e.g., factoring is hard) or generic (e.g., there exist one-way functions or there exist trapdoor permutations). The minimal generic assumption is the existence of one-way functions. This assumption implies, for example, that pseudorandom generators and private-key encryption systems exist [41] and digital signatures exist [55]. However, many other tasks are not known to follow from one-way functions. Impagliazzo and Rudich [44] showed that using blackbox reductions one cannot construct oblivious-transfer protocols based on one-way functions.

The next result of Rudich [56] shows how to construct oblivious-transfer protocols based on one-way functions and an efficient secret-sharing scheme for $\mathcal{A}_{\text{ham}}$. By Theorem 6, we cannot hope for a perfect secret-sharing scheme for $\mathcal{A}_{\text{ham}}$. However, if one can construct computational secret-sharing schemes realizing $\mathcal{A}_{\text{ham}}$ based on one-way functions, then we get that one-way functions imply oblivious-transfer protocols. This will solve a major open problem in cryptography, i.e., using Impagliazzo's terminology [43], it will prove that Minicrypt = Cryptomania. As Rudich's result uses a non-blackbox reduction, such construction bypasses the impossibility result of [44].

Preliminaries. In this survey we will not define computational secret-sharing schemes. This definition can be found in [12]. In such schemes we require that the sharing and reconstruction are done in polynomial-time in the secret length and the number of parties in the access structure. Furthermore, we require that a polynomial-time adversary controlling of an unauthorized set cannot distinguish between shares of one secret and shares of another secret.

Rudich considers schemes for $\mathcal{A}_{\text{ham}}$, where the requirement on efficient reconstruction is quite weak: any authorized subset E can efficiently reconstruct the secret given that the set knows the Hamiltonian cycle in E. Thus, this weaker requirement avoids problems arising from the NP-completeness of the Hamiltonian problem.

Next, we recall the notion of 1-out-of-2 oblivious transfer [53, 35]. This is a two party protocol between two parties, a sender holding two bits b_0, b_1 and a receiver holding an index $i \in \{0, 1\}$. At the end of the protocol, the receiver should hold b_i without gaining any knowledge on the other bit b_{1-i}. The sender should not be able to deduce any information on i. Intuitively, the sender sends exactly one bit to the receiver, however, it is oblivious to which bit it sends. As in Section 4, we consider honest-but-curious parties. As the result of [44] already applies to this setting, constructing oblivious-transfer protocols for honest-but-curious parties is already interesting. Furthermore, by a transformation of [39], any such protocol can be transformed into a protocol secure against malicious parties assuming that one-way functions exist.

We are ready to state and prove Rudich's result.

Theorem 7 ([56]). *If one-way functions exist and an efficient secret-sharing scheme for the Hamiltonian access structure $\mathcal{A}_{\text{ham}}$ exists then oblivious-transfer protocols exist.*

Proof. Let Gen be a pseudorandom generator stretching ℓ bits to 2ℓ bits. By [41], if one-way functions exist, then such Gen exists. Define the language $L_{\text{Gen}} = \{y : \exists_x \text{Gen}(x) = y\}$. Clearly, $L_{\text{Gen}} \in NP$. Let f be a polynomial-time reduction from L_{Gen} to Hamiltonian, that is, f can be computed in polynomial time and $y \in L_{\text{Gen}}$ iff $G = f(y) \in$ Hamiltonian. Such f exists with the property that a witness to y can be efficiently translated to a witness to $G = f(x)$, that is, given $y \in L_{\text{Gen}}$, a witness x for it, that is, $\text{Gen}(x) = y$, and $G = f(x)$, one can find in polynomial time a Hamiltonian cycle in G. The next protocol is an oblivious-transfer protocol (for honest-but-curious parties):

Receiver's input: $i \in \{0,1\}$ and security parameter 1^ℓ.
Sender's input: b_0, b_1 and security parameter 1^ℓ.
Instructions for the receiver:
- Choose at random $x_1 \in \{0,1\}^\ell$ and compute $y_1 = \text{Gen}(x_1)$.
- Choose at random $y_0 \in \{0,1\}^{2\ell}$.
- Compute $G_\sigma = f(y_\sigma)$ for $\sigma \in \{0,1\}$.
- If $i = 0$ send G_1, G_0 to the sender, else send G_0, G_1 to the sender.

Instructions for the sender:
- Let $H_0 = (V_0, E_0), H_1 = (V_1, E_1)$ be the graphs that the receiver sends.
- For $j \in \{0,1\}$, share the bit b_j using the scheme for the Hamiltonian access structure $\mathcal{A}_{\text{ham}}$ for the complete graph with $|V_j|$ vertices, and send the shares of the parties corresponding to the edges in E_j to the receiver.

Instructions for the receiver: Compute a Hamiltonian cycle in G_1 from x_1 and y_1, and reconstruct b_i from the shares of this cycle for the graph $H_i = G_1$.

The privacy of the receiver is protected since the sender cannot efficiently distinguish between a string sampled according to the uniform distribution in $\{0,1\}^{2\ell}$ and an output of the generator on a string sampled uniformly in $\{0,1\}^\ell$. In particular, the sender cannot efficiently distinguish between the output of the reduction f on two such strings.

The privacy of the sender is protected against an honest-but-curious receiver since with probability at least $1 - 1/2^\ell$ the string y_0 is not in the range of Gen, thus, G_{1-i} has no Hamiltonian cycle, that is, E_i is an unauthorized set. In this case, the secret b_{1-i} cannot be efficiently computed from the shares of E_{1-i}. □

If we hope to construct an oblivious-transfer protocol using the approach of Theorem 7, then we should construct an efficient computational scheme for the Hamiltonian access structure based on the assumption that one-way functions exist. For feasibility purposes it would be interesting to construct a computational secret-sharing scheme for Hamiltonicity based on stronger cryptographic assumptions, e.g., that trapdoor permutations exist.

7 Summary and Open Problems

In this survey we consider secret-sharing schemes, a basic tool in cryptography. We show several constructions of secret-sharing schemes, starting from the

scheme of [45]. We then described its generalization by [14], showing that if an access structure can be described by a small monotone formula, then it has an efficient secret-sharing scheme. We also showed the construction of secret-sharing schemes from monotone span programs [21, 46]. Monotone span programs are equivalent to linear secret-sharing schemes and are equivalent to schemes where the reconstruction is linear [3]. 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 [14]. Furthermore, there are functions that have small monotone span programs and do not have small monotone formulae [2], thus, this is a strict generalization. Finally, we presented the multi-linear construction of secret-sharing schemes.

All the constructions presented in Section 3 are linear over a finite field (some of the schemes work also over finite groups, e.g., the scheme of Benaloh and Leichter). The linearity of a scheme is important in many applications, as we demonstrated in Section 4 for the construction of secure multiparty protocols for general functions. Thus, it is interesting to understand the access structures that have efficient linear secret-sharing schemes. The access structures that can efficiently be realized by linear and multi-linear secret-sharing scheme are characterized by functions that have polynomial size monotone span programs, or, more generally, multi-target monotone span programs. We would like to consider the class of access structures that can be realized by linear secret-sharing schemes with polynomial share length. As this discussion is asymptotic, we consider a sequence of access structures $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$, where $\mathcal{A}_n$ has n parties. As linear algebra can be computed in NC (informally, NC is the class of problems that can be solved by parallel algorithms with polynomially many processors and poly-logarithmic running time), every sequence of access structures that has efficient linear secret-sharing schemes can be recognized by NC algorithms. For example, if $P \neq NC$, then access structures recognized by monotone P-complete problems do not have efficient linear secret-sharing schemes.

The limitations of linear secret-sharing schemes raise the question if there are non-linear secret-sharing schemes. Beimel and Ishai [6] have constructed non-linear schemes for access structures that are not known to be in P (e.g., for an access structure related to the quadratic residuosity problem over $N = pq$). Thus, non-linear schemes are probably stronger than linear schemes. Furthermore, Beimel and Ishai defined quasi-linear schemes, which are compositions of linear schemes over different fields. Beimel and Weinreb [10] showed, without any assumptions, that quasi-linear schemes are stronger than linear schemes, that is, there exists an access structure that has quasi-linear schemes with constant information ratio while every linear secret-sharing scheme realizing this access structure has super-polynomial information ratio. However, Beimel and Ishai [6] proved that if an access structure has efficient quasi-linear scheme, then it can be recognized by an NC algorithm. Thus, also the class of access structures realized by efficient quasi-linear schemes is limited.

Another non-linear construction of secret-sharing schemes is an unpublished result of Yao [69] (see also [67]). Yao showed that if an access structure can be

described by a small monotone *circuit*, then it has an efficient computational secret-sharing scheme. This generalizes the results of [14] showing that if an access structure can be described by a small monotone *formula*, then it has an efficient perfect secret-sharing scheme. We will not describe the construction of Yao in this survey.

An additional topic that we will not cover in this survey is ideal secret-sharing schemes. By Lemma 2, the size of the share of each party is at least the size of the secret. An ideal secret-sharing scheme is a scheme in which the size of the share of each party is exactly the size of the secret. For example, Shamir's scheme [58] is ideal. An access structure is ideal if it has an ideal scheme over some finite domain of secrets. For example, threshold access structures are ideal, while the access structure $\sqcap$ described in Example 5 is not ideal. Brickell [21] considered ideal schemes and constructed ideal schemes for some access structures, i.e., for hierarchical access structures. Brickell and Davenport [22] showed an interesting connection between ideal access structures and matroids, that is,

- 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.

Following this work, many works have constructed ideal schemes, and have studied ideal access structures and matroids. For example, Martí-Farré and Padró [49] 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).

7.1 Open Problems

The most important open problem regarding secret-sharing schemes is settling Conjecture 1. That is,

Question 1. *Prove (or disprove) that there exists an access structure such that the information ratio of every secret-sharing scheme realizing it is* $2^{\Omega(n)}$.

A weaker version of this question is the following:

Question 2. *Prove (or disprove) that there exists an access structure such that the information ratio of every secret-sharing scheme realizing it with domain of secrets* $\{0,1\}$ *is super-polynomial in* n.

The above two questions are open even for non-explicit access structures. For linear schemes, we have super-polynomial lower bounds for explicit access structures. By counting arguments, for every n for most access structures with n parties the size of shares in every linear secret-sharing scheme realizing them is $2^{\Omega(n)}$. It is open to prove such lower bound for explicit access structures.

Question 3. *Prove that there exists an* explicit *access structure such that the information ratio of every linear secret-sharing scheme realizing it is* $2^{\Omega(n)}$.

In this survey, we describe linear and multi-linear secret-sharing schemes. It is known that multi-linear schemes are more efficient than linear schemes for small access structures, e.g., [62]. However, the possible improvement by using multi-linear schemes compared to linear schemes is open.

Question 4. *Given an access structure, what is the biggest gap between best information ratio of multi-linear schemes realizing the access structure compared to the best information ratio of linear schemes realizing the access structure?*

There are interesting access structures that we do not know if they have efficient schemes. 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 [69] it has an efficient computational scheme. It is not known if this access structure can be described by a small monotone span program and it is open if it has an efficient perfect scheme. In [9], it was proved that every monotone span program accepting the directed connectivity access structure has size $\Omega(n^{3/2})$. In comparison, the *undirected connectivity* access structure has an efficient perfect scheme [15] (see Section 3.2).

The second access structure that we do not know if it has an efficient scheme is the *perfect matching* access structure. 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. It is not even known if this access structure has an efficient computational scheme as every monotone circuit for perfect matching has super-polynomial size. We remark that an efficient scheme for this access structure implies an efficient scheme for the directed connectivity access structure.

The third interesting family of access structures is *weighted threshold* access structures. In such an access structure each party has a weight and there is some threshold. A set of parties is authorized if the sum of the weights of the parties in the set is bigger than the threshold. For these access structures there is an efficient computational scheme [11] and a perfect scheme with $n^{O(\log n)}$ long shares. It is open if these access structures have a perfect scheme with polynomial shares. Furthermore, it is open if they can be described by polynomial size monotone formulae.

Acknowledgment

I would like to thank Benny Chor for introducing me to the field of secret-sharing schemes and guiding me in the early stages of my career. I would also like to thank my co-authors in papers related to secret sharing: Mike Burmester, Yvo Desmedt, Matt Franklin, Anna Gál, Yuval Ishai, Eyal Kushilevitz, Noam Livne, Ilan Orlov, Carles Padró, Anat Paskin, Mike Paterson, Tamir Tassa, and Enav Weinreb. I learned a lot from working with them. Finally, thanks to Moni Naor for telling me about the unpublished results of Rudich presented in Section 6.

References

1. N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3:289–304, 1992.
2. L. Babai, A. Gál, and A. Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.
3. A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Technion, 1996. www.cs.bgu.ac.il/ beimel/pub.html.
4. A. Beimel and B. Chor. Universally ideal secret sharing schemes. *IEEE Trans. on Information Theory*, 40(3):786–794, 1994.
5. A. Beimel, A. Gál, and M. Paterson. Lower bounds for monotone span programs. *Computational Complexity*, 6(1):29–45, 1997. Conference version: FOCS '95.
6. A. Beimel and Y. Ishai. On the power of nonlinear secret-sharing. *SIAM J. on Discrete Mathematics*, 19(1):258–280, 2005.
7. A. Beimel, N. Livne, and C. Padró. Matroids can be far from ideal secret sharing. In R. Canetti, editor, *Proc. of the Fifth Theory of Cryptography Conference – TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 194–212, 2008.
8. A. Beimel and I. Orlov. Secret sharing and non-shannon information inequalities. *IEEE Trans. on Information Theory*, 2011. Accepted for publication. Preliminary version in TCC 2009, *LNCS* vol, 5444:539–557, 2009.
9. A. Beimel and A. Paskin. On linear secret sharing for connectivity in directed graphs. In R. Ostrovsky, R. De Prisco, and I. Visconti, editors, *Proc. of the Sixth Conference on Security and Cryptography for Networks*, volume 5229 of *Lecture Notes in Computer Science*, pages 172–184. Springer-Verlag, 2008.
10. A. Beimel and E. Weinreb. Separating the power of monotone span programs over different fields. *SIAM J. on Computing*, 34(5):1196–1215, 2005.
11. A. Beimel and E. Weinreb. Monotone circuits for monotone weighted threshold functions. *Inform. Process. Lett.*, 97(1):12–18, 2006. Conference version: *Proc. of 20th Annu. IEEE Conf. on Computational Complexity*, pages 67-75, 2005.
12. M. Bellare and P. Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proc. of the 14th ACM conference on Computer and communications security*, pages 172–184, 2007.
13. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computations. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 1–10, 1988.
14. J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer-Verlag, 1990.
15. J. Benaloh and S. Rudich. Private communication, 1989.
16. M. Bertilsson and I. Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology – AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 67–79. Springer-Verlag, 1993.
17. G. R. Blakley. Safeguarding cryptographic keys. In R. E. Merwin, J. T. Zanca, and M. Smith, editors, *Proc. of the 1979 AFIPS National Computer Conference*, volume 48 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.
18. C. Blundo, A. De Santis, R. De Simone, and U. Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Designs, Codes and Cryptography*, 11(2):107–122, 1997.

19. C. Blundo, A. De Santis, L. Gargano, and U. Vaccaro. On the information rate of secret sharing schemes. *Theoretical Computer Science*, 154(2):283–306, 1996.
20. C. Blundo, A. De Santis, and U. Vaccaro. On secret sharing schemes. *Inform. Process. Lett.*, 65(1):25–32, 1998.
21. E. F. Brickell. Some ideal secret sharing schemes. *Journal of Combin. Math. and Combin. Comput.*, 6:105–113, 1989.
22. E. F. Brickell and D. M. Davenport. On the classification of ideal secret sharing schemes. *J. of Cryptology*, 4(73):123–134, 1991.
23. R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro. On the size of shares for secret sharing schemes. *J. of Cryptology*, 6(3):157–168, 1993.
24. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 11–19, 1988.
25. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *Proc. of the 26th IEEE Symp. on Foundations of Computer Science*, pages 383–395, 1985.
26. B. Chor and E. Kushilevitz. Secret sharing over infinite domains. *J. of Cryptology*, 6(2):87–96, 1993.
27. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
28. R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer-Verlag, 2000.
29. L. Csirmaz. The size of a share must be large. In A. De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 13–22. Springer-Verlag, 1995. Journal version in: *J. of Cryptology*, 10(4):223–231, 1997.
30. L. Csirmaz. The dealer's random bits in perfect secret sharing schemes. *Studia Sci. Math. Hungar.*, 32(3–4):429–437, 1996.
31. Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469. Springer-Verlag, 1992.
32. M. van Dijk. A linear construction of perfect secret sharing schemes. In A. De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 23–34. Springer-Verlag, 1995.
33. M. van Dijk. On the information rate of perfect secret sharing schemes. *Designs, Codes and Cryptography*, 6:143–169, 1995.
34. M. van Dijk, T. Kevenaar, G.-J. Schrijen, and P. Tuyls. Improved constructions of secret sharing schemes by applying $(\lambda], \omega)$-decompositions. *Inform. Process. Lett.*, 99(4):154 – 157, 2006.
35. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *CACM*, 28(6):637–647, 1985.
36. A. Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2002.
37. A. Gál and P. Pudlák. Monotone complexity and the rank of matrices. *Inform. Process. Lett.*, 87:321–326, 2003.
38. R. Gennaro, M. O. Rabin, and T. Rabin. Simplified vss and fact-track multiparty computations with applications to threshold cryptography. In *Proc. of the 17th ACM Symp. on Principles of Distributed Computing*, pages 101–111, 1998.

39. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. of the 19th ACM Symp. on the Theory of Computing*, pages 218–229, 1987.
40. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.
41. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudo-random generator from any one-way function. *SIAM J. on Computing*, 28(4):1364–1396, 1999.
42. M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. of Cryptology*, 13(1):31–60, 2000.
43. R. Impagliazzo. A personal view of average-case complexity. In *Proc. of the 10th IEEE Structure in Complexity Theory*, pages 134–147, 1995.
44. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. of the 21st ACM Symp. on the Theory of Computing*, pages 44–61, 1989.
45. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. of the IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1):15-20, 1993.
46. M. Karchmer and A. Wigderson. On span programs. In *Proc. of the 8th IEEE Structure in Complexity Theory*, pages 102–111, 1993.
47. E. D. Karnin, J. W. Greene, and M. E. Hellman. On secret sharing systems. *IEEE Trans. on Information Theory*, 29(1):35–41, 1983.
48. E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
49. J. Martí-Farré and C. Padró. On secret sharing schemes, matroids and polymatroids. In S. Vadhan, editor, *Proc. of the Fourth Theory of Cryptography Conference – TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 253–272. Springer-Verlag, 2007.
50. F. Matúš. Infinitely many information inequalities. In *IEEE International Symposium on Information Theory 2007*, pages 41–44, 2007.
51. J. R. Metcalf-Burton. Improved upper bounds for the information rates of the secret sharing schemes induced by the Vamos matroid. Technical Report abs/0809.3010, CoRR, 2008.
52. M. Naor and A. Wool. Access control and signatures via quorum secret sharing. *IEEE Transactions on Parallel and Distributed Systems*, 9(1):909–922, 1998.
53. M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981. Available online in the Cryptology ePrint Archive, Report 2005/187, eprint.iacr.org/2005/187.
54. M. O. Rabin. Randomized Byzantine generals. In *Proc. of the 24th IEEE Symp. on Foundations of Computer Science*, pages 403–409, 1983.
55. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. of the 22nd ACM Symp. on the Theory of Computing*, pages 387–394, 1990.
56. S. Rudich. Private communication (via M. Naor), 1989.
57. A. Sahai and B. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 457–473. Springer-Verlag, 2005.
58. A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
59. B. Shankar, K. Srinathan, and C. Pandu Rangan. Alternative protocols for generalized oblivious transfer. In S. Rao, M. Chatterjee, P. Jayanti, C. S. Murthy,

and S. K. Saha, editors, *Proc. of ICDCN 2008*, volume 4904 of *Lecture Notes in Computer Science*, pages 304–309. Springer-Verlag, 2008.
60. G. J. Simmons. How to (really) share a secret. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 390–448. Springer-Verlag, 1990.
61. G. J. Simmons, W. Jackson, and K. M. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.
62. J. Simonis and A. Ashikhmin. Almost affine codes. *Designs, Codes and Cryptography*, 14(2):179–197, 1998.
63. D. R. Stinson. Decomposition construction for secret sharing schemes. *IEEE Trans. on Information Theory*, 40(1):118–125, 1994.
64. T. Tassa. Hierarchical threshold secret sharing. In M. Naor, editor, *Proc. of the First Theory of Cryptography Conference – TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 473–490. Springer-Verlag, 2004.
65. T. Tassa. Generalized oblivious transfer by secret sharing. *Designs, Codes and Cryptography*, 58, 2011.
66. T. Tassa and N. Dyn. Multipartite secret sharing by bivariate interpolation. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proc. of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 288–299. Springer-Verlag, 2006.
67. V. Vinod, A. Narayanan, K. Srinathan, C. Pandu Rangan, and K. Kim. On the power of computational secret sharing. In T. Johansson and S. Maitra, editors, *Indocrypt 2003*, volume 2904 of *Lecture Notes in Computer Science*, pages 162–176. Springer-Verlag, 2003.
68. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Technical Report 2008/290, Cryptology ePrint Archive, 2008. http://eprint.iacr.org/.
69. A. C. Yao. Unpublished manuscript, 1989. Presented at Oberwolfach and DIMACS workshops.
70. R. W. Yeung. *Information Theory and Network Coding*. Springer, 2008.
71. Z. Zhang and R. W. Yeung. On characterization of entropy function via information inequalities. *IEEE Trans. on Information Theory*, 44(4):1440–1452, 1998.