Secret-Sharing Schemes for General and Uniform Access Structures

Benny Applebaum[1], Amos Beimel[2], Oriol Farràs[3], Oded Nir[1], and Naty Peter[2]

[1] Tel Aviv University, Tel Aviv, Israel
[2] Ben-Gurion University of the Negev, Be'er-Sheva, Israel
[3] Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
benny.applebaum@gmail.com, amos.beimel@gmail.com, oriol.farras@urv.cat, odednir123@gmail.com, naty@post.bgu.ac.il

Abstract. A secret-sharing scheme allows some authorized sets of parties to reconstruct a secret; the collection of authorized sets is called the access structure. For over 30 years, it was known that any (monotone) collection of authorized sets can be realized by a secret-sharing scheme whose shares are of size $2^{n-o(n)}$ and until recently no better scheme was known. In a recent breakthrough, Liu and Vaikuntanathan (STOC 2018) have reduced the share size to $O(2^{0.994n})$. Our first contribution is improving the exponent of secret sharing down to 0.892. For the special case of linear secret-sharing schemes, we get an exponent of 0.942 (compared to 0.999 of Liu and Vaikuntanathan).

Motivated by the construction of Liu and Vaikuntanathan, we study secret-sharing schemes for uniform access structures. An access structure is k-uniform if all sets of size larger than k are authorized, all sets of size smaller than k are unauthorized, and each set of size k can be either authorized or unauthorized. The construction of Liu and Vaikuntanathan starts from protocols for conditional disclosure of secrets, constructs secret-sharing schemes for uniform access structures from them, and combines these schemes in order to obtain secret-sharing schemes for general access structures. Our second contribution in this paper is constructions of secret-sharing schemes for uniform access structures. We achieve the following results:

- A secret-sharing scheme for k-uniform access structures for large secrets in which the share size is $O(k^2)$ times the size of the secret.
- A linear secret-sharing scheme for k-uniform access structures for a binary secret in which the share size is $\tilde{O}(2^{h(k/n)n/2})$ (where h is the binary entropy function). By counting arguments, this construction is optimal (up to polynomial factors).
- A secret-sharing scheme for k-uniform access structures for a binary secret in which the share size is $kn \cdot 2^{\tilde{O}(\sqrt{k \log n})}$.

Our third contribution is a construction of ad-hoc PSM protocols, i.e., PSM protocols in which only a subset of the parties will compute a function on their inputs. This result is based on ideas we used in the construction of secret-sharing schemes for k-uniform access structures for a binary secret.

1 Introduction

A secret-sharing scheme is a method in which a dealer that holds a secret information (e.g., a password or private medical data) can store it on a distributed system, i.e., among a set of parties, such that only some predefined authorized sets of parties can reconstruct the secret. The process of storing the secret information is called secret sharing and the collections of authorized sets of parties are called access structures. Interestingly, secret-sharing schemes are nowadays used in numerous applications (in addition to their obvious usage for secure storage), e.g., they are used for secure multiparty computation [15, 19], threshold cryptography [24], access control [38], attribute-based encryption [29, 44], and oblivious transfer [40, 43]. The original and most important secret-sharing schemes, introduced by Blakley [18] and Shamir [39], are threshold secret-sharing schemes, in which the authorized sets are all the sets whose size is larger than some threshold.

Secret-sharing schemes for general access structures were introduced in [32] more than 30 years ago. However, we are far from understanding constraints on the share size of these schemes. In the original constructions of secret-sharing schemes in [32], the share size of each party is $2^{n-O(\log n)}$. New constructions of secret-sharing schemes followed, e.g., [16, 33, 17]; however, the share size of each party in these schemes remains $2^{n-O(\log n)}$. In a recent breakthrough, Liu and Vaikuntanathan [34] (building on [36]) showed, for the first time, that it is possible to construct secret-sharing schemes in which the share size of each party is $O(2^{cn})$ with an exponent c strictly smaller than 1. In particular, they showed that every access structure can be realized with an exponent of $\mathbf{S}_{\mathrm{LV}} = 0.994$. Moreover, they showed that every access structure can be realized by a linear secret-sharing scheme with an exponent of 0.994 (a scheme is linear if each share can be written as a linear combination of the secret and some global random field elements; see Section 2 for a formal definition). On the negative size, the best lower bound on the total share size required for sharing a secret for some access structure is $\Omega(n^2/\log n)$ [22, 23]. Thus, there is a huge gap between the known upper and lower bounds.

1.1 Our Results

Our first result is an improvement of the secret-sharing exponent of general access structure. In Section 3, we prove the following theorem.

Theorem 1.1. *Every access structure over n parties can be realized by a secret-sharing scheme with a total share size of $2^{0.8916n+o(n)}$ and by a linear secret-sharing scheme with a total share size of $2^{0.942n+o(n)}$.*

In a nutshell, the construction of [34] together with combinatorial covering designs are being used to establish a recursive construction, which eventually leads to the improved bounds.

We next construct secret-sharing schemes for uniform access structures. An access structure is k-uniform if all sets of size larger than k are authorized, all

sets of size smaller than k are unauthorized, and every set of size k can be either authorized or unauthorized. Our second contribution is on the construction of secret-sharing schemes for uniform access structures. The motivation for studying uniform access structures is twofold. First, they are related to protocols for conditional disclosure of secrets (CDS), a primitive introduced by Gertner et al. [28]. By various transformations [11, 12, 34, 2, 13], CDS protocols imply secret-sharing schemes for uniform access structures. Furthermore, as shown in [34], CDS protocols and secret-sharing schemes for uniform access structures are a powerful primitives to construct secret-sharing schemes for general access structures. Thus, improvements on secret-sharing schemes for uniform access structures can lead to better constructions of secret-sharing schemes for general access structures. Second, as advocated in [2], uniform access structures should be studied as they are a useful scaled-down version of general access structures. Studying uniform access structures can shed light on the share size required for general access structures, which is a major open problem.

Three regimes of secret-sharing schemes for uniform access structures have been studied. The first regime is the obvious one of secret-sharing schemes with short secrets (e.g., a binary secret). The second regime is secret-sharing schemes with long secrets. Surprisingly, there are secret-sharing for this regime that are much more efficient than schemes with short secrets [2]. The third regime is *linear* secret-sharing schemes with short secrets. Linear secret-sharing schemes are schemes where the sharing of the secret is done using a linear transformation; such schemes are interesting since in many applications linearity is required, e.g., in the construction of secure multiparty computation protocols in [21] and in the constructions of Attrapadung [7] and Wee [45] of public-key (multi-user) attribute-based encryption.

In this paper we improve the constructions of secret-sharing schemes for uniform access structures in these three regimes. We describe our results according to the order that they appear in the paper.

Long secrets. In Section 4, we construct secret-sharing schemes for n-party k-uniform access structures for large secrets, i.e., secrets of size at least 2^{n^k}. Previously, the share size in the best constructions for such schemes was either e^k times the length of the secret [2] or n times the length of the secret (implied by the CDS protocol of [2] and a transformation of [12]). We show a construction in which the share size is $O(k^2)$ times the size of the secret. For this construction, we use the CDS protocol of [2] with k^2 parties (in contrast to [2], which uses it with k parties) with an appropriate k^2-input function. Combined with the results of [12], we get a share size which is at most $\min(k^2, n)$-times larger than the secret size.

Linear schemes. In Section 5, we design a linear secret-sharing scheme for k-uniform access structures for a binary secret in which the share size is $\tilde{O}(2^{h(k/n)n/2})$ (where h is the binary entropy function). By counting arguments, our construction is optimal (up to polynomial factors). Previously, the best construction was implied by the CDS protocols of [13, 36] and had share size $\tilde{O}(2^{n/2})$.

Our construction is inspired by a linear 2-party CDS protocol of [27] and the linear k-party CDS protocols of [13]. We use the ideas of these CDS protocols to design a linear secret-sharing schemes for balanced k-uniform access structures (where there is a set B of $n/2$ parties such that any minimal authorized set of size k in the access structure contains exactly $k/2$ parties from B). Using a probabilistic argument, we show that every k-uniform access structure can be written as a union of $O(n^{3/2})$ balanced access structures, thus, we can share the secret independently for each balanced access structure in the union.

Short secrets. In Section 6, we describe a secret-sharing scheme for k-uniform access structures for a binary secret in which the share size is $kn \cdot 2^{\tilde{O}(\sqrt{k \log n})}$. Previously, the best share size in a secret-sharing scheme realizing such access structures was $\min\left\{2^{O(k)+\tilde{O}(\sqrt{k \log n})}, 2^{\tilde{O}(\sqrt{n})}\right\}$, (by combining the results of [36] with those of [2] and [12] respectively). We note that when k is very small (e.g., constant), the scheme based on [36] and on the work of Applebaum and Arkis [2] outperforms our scheme. To achieve our result we define a new transformation from a k-party CDS protocol to secret-sharing schemes for k-uniform access structures. The idea of this transformation is that the shares of the parties contain the messages in a CDS protocol of an appropriate function. The difficulty is how to ensure that parties of an unauthorized set of size k cannot obtain two messages of the same party in the CDS protocol (otherwise, the privacy of the CDS protocol can be violated). We achieve this goal by appropriately sharing the CDS messages among the parties.

Ad-hoc PSM. We also study private simultaneous messages (PSM) protocols, which is a minimal model of secure multiparty computation protocols. In a PSM protocol there are k parties and a referee; each party holds a private input x_i and sends one message to the referee without seeing the messages of the other parties. The referee should learn the output of a pre-defined function $f(x_1, \ldots, x_n)$ without learning any additional information on the inputs. We use the ideas of the last transformation to construct ad-hoc PSM protocols (a primitive introduced in [14]), i.e., PSM protocols in which only a subset of the parties will compute a function on their inputs. We show that if a function f has a k-party PSM protocol with complexity C, then it has a k-out-of-n ad-hoc PSM protocol with complexity $O(knC)$.

1.2 Related Work

Constructions of secret-sharing schemes. Shamir [39] and Blakley [18] showed that threshold access structures can be realized by linear secret-sharing schemes, in which the size of every share is the maximum between the $\log n$ and the secret size. Ito, Saito, and Nishizeki constructed secret-sharing schemes for general access structures in which the share size is proportional either to the DNF or CNF representation of the access structure. Benaloh and Leichter [16] showed that access structures that can be described by small monotone formulas can be

realized by efficient secret-sharing schemes. Later, Karchmer and Wigderson [33] showed that access structures that can be described by small monotone span programs can also be realized by efficient secret-sharing schemes. Bertilsson and Ingemarsson [17] presented multi-linear secret-sharing schemes for general access structures. All the above schemes have share size $2^{n-O(\log n)}$. This was recently improved in [34] (as we have already explained).

Secret-sharing schemes for uniform access structures. Secret-sharing schemes for 2-uniform access structures were first introduced by Sun and Shieh [42]. Such schemes are called schemes for prohibited or forbidden graphs. 2-uniform access structures were studied in many papers, such as [11, 27, 10, 9, 35, 36, 3, 2, 13]. Beimel et al. [11] proved that every 2-uniform access structure can be realized by a (non-linear) secret-sharing scheme in which the share size of every party is $O(n^{1/2})$. Later, Gay et al. [27] presented linear secret-sharing schemes for such access structures with the same share size. Liu et al. [35] constructed non-linear secret-sharing scheme for 2-uniform access structures in which the share size of every party is $2^{O(\sqrt{\log n \log\log n})} = n^{o(1)}$. The notion of k-uniform access structures was explicitly introduced by [2, 12] and was implicit in the work of [34]. By combining the CDS protocol of [36] and transformations of [2, 12], one can obtain that every k-uniform access structure can be realized by a secret-sharing scheme in which the share size of every party is $\min\left\{2^{O(k)+\tilde{O}(\sqrt{k\log n})}, 2^{\tilde{O}(\sqrt{n})}\right\}$. Applebaum and Arkis [2] (extending the work of Applebaum et al. [3]) showed a secret-sharing scheme for k-uniform access structures for long secrets, in which the share size of every party is $O(e^k)$ times the secret size (for long secrets). Recently, Beimel and Peter [13] proved that every k-uniform access structure can be realized by a linear secret-sharing scheme in which the share of every party is $\min\left\{(O(n/k))^{(k-1)/2}, O(n\cdot 2^{n/2})\right\}$.

Conditional disclosure of secrets (CDS) Protocols. Our constructions, described in Section 1.1, start from CDS protocols and transform them to secret-sharing schemes. In a conditional disclosure of secrets protocol, there are k parties and a referee; each party holds a private input, a common secret, and a common random string. The referee holds all private inputs but, prior to the protocol, it does not know neither the secret nor the random string. The goal of the protocol is that the referee will learn the secret if and only if the inputs of the parties satisfy some pre-defined condition (e.g., all inputs are equal). The challenge is that the communication model is minimal – each party sends one message to the referee, without seeing neither the inputs of the other parties nor their inputs.

CDS protocols were introduced by Gertner et al. [28], who presented a linear k-party CDS protocol for k-input functions $f : [N]^k \to \{0,1\}$ with message size $O(N^k)$. CDS protocols are used in the constructions of many cryptographic protocols, for example, symmetrically-private information retrieval protocols [28], attribute based encryption [27, 7, 45], and priced oblivious transfer [1].

CDS protocols have been studied in many papers [31, 27, 10, 3, 9, 35, 2, 36, 12, 13]. In the last few years there were dramatic improvements in the message size

of CDS protocols. For a function $f : [N]^k \to \{0,1\}$, the message size in the best known CDS protocols is as follows: (1) For a binary secret, the message size is $2^{\tilde{O}(\sqrt{k \log N})}$ [36]. (2) For long secrets (of size at least 2^{N^k-1}), the message size is 4 times the size of the secret [2]. (3) For a binary secret, there is a linear CDS protocol with message size $O(N^{(k-1)/2})$ [36, 13]. The best known lower-bound for general CDS protocol is $\Omega(\log N)$ [3–5]

Private simultaneous messages (PSM) Protocols. The model of k-party PSM protocols for k-input functions $f : [N]^k \to \{0,1\}$ was first introduced by Feige et al. [26], for $k = 2$, and was generalized to any k in [26, 30]. In [26], it was shown that every 2-input function has a 2-party PSM protocol with message size $O(N)$. Beimel et al. [11] improved this result by presenting a 2-party PSM protocol with messages size $O(N^{1/2})$. The best known lower bound for such 2-party PSM protocol is $3 \log N - O(\log \log N)$ [26, 4]. It was shown by Beimel et al. [12] that there exists a k-party PSM protocol with message size $O(k^3 \cdot N^{k/2})$.

Ad-hoc PSM protocols were presented by Beimel et al. in [14]. They showed that if there is a k-party PSM protocol for a symmetric function f with message size C, then there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(k^3 \cdot e^k \cdot \log n \cdot C)$. Thus, by the PSM protocol of [12], there is a k-out-of-n ad-hoc PSM protocol for every symmetric function with message size $O(k^6 \cdot e^k N^{k/2} \cdot \log n)$. In [14], they also showed that if there is a n-party PSM protocol for a function f' related to f, with message size C, then there is a k-out-of-n ad-hoc PSM protocol for f with message size $n \cdot C$. This construction implies, in particular, that ad-hoc PSM protocols with poly(n)-communication exist for NC1 and different classes of log-space computation.

2 Preliminaries

Secret-Sharing Schemes. We present the definition of secret-sharing schemes, similar to [8, 20].

Definition 2.1 (Access Structures). *Let $P = \{P_1, \ldots, P_n\}$ be a set of parties. A collection $\Gamma \subseteq 2^P$ is* monotone *if $B \in \Gamma$ and $B \subseteq C$ imply that $C \in \Gamma$. An* access structure *is a monotone collection $\Gamma \subseteq 2^P$ of non-empty subsets of P. Sets in Γ are called* authorized*, and sets not in Γ are called* unauthorized*. The family of minimal authorized subsets is denoted by $\min \Gamma$. We represent a subset of parties $A \subseteq P$ by its characteristic string $x_A = (x_1, \ldots, x_k) \in \{0,1\}^n$, where for every $j \in [n]$ it holds that $x_j = 1$ if and only if $P_j \in A$. For an access structure Γ, we define the function $f_\Gamma : \{0,1\}^n \to \{0,1\}$, where for every subset of parties $A \subseteq P$, it holds that $f_\Gamma(x_A) = 1$ if and only if $A \in \Gamma$.*

Definition 2.2 (Secret-Sharing Schemes). *A* secret-sharing scheme *with domain of secrets S is a pair $\Sigma = \langle \Pi, \mu \rangle$, where μ is a probability distribution on some finite set R called the set of random strings and Π is a mapping from $S \times R$ to a set of n-tuples $S_1 \times S_2 \times \cdots \times S_n$, where S_j is called the* domain of shares *of P_j. A dealer distributes a secret $s \in S$ according to Σ by first sampling a random*

string $r \in R$ according to μ, computing a vector of shares $\Pi(s,r) = (s_1, \ldots, s_n)$, *and privately communicating each share s_j to party P_j. For a set $A \subseteq P$, we denote $\Pi_A(s,r)$ as the restriction of $\Pi(s,r)$ to its A-entries (i.e., the shares of the parties in A).*

Given a secret-sharing scheme Σ, define the size *of the secret as* $\log|S|$, *the* share size *of party P_j as* $\log|S_j|$, *the* max share size *as* $\max_{1\leq j\leq n}\{\log|S_j|\}$, *and the* total share size *as* $\sum_{j=1}^{n}\log|S_j|$.

Let S be a finite set of secrets, where $|S| \geq 2$. A secret-sharing scheme $\Sigma = \langle \Pi, \mu \rangle$ with domain of secrets S realizes an access structure Γ if the following two requirements hold:

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

PRIVACY. *Every unauthorized set cannot learn anything about the secret from its shares. Formally, there exists a randomized function* SIM, *called the* simulator, *such that for any set $T = \{P_{i_1}, \ldots, P_{i_{|T|}}\} \notin \Gamma$, every secret $s \in S$, and every vector of shares $(s_{i_1}, \ldots, s_{i_{|T|}}) \in S_{i_1} \times \cdots \times S_{i_{|T|}}$,*

$$\Pr[\,\mathrm{SIM}(T) = (s_{i_1}, \ldots, s_{i_{|T|}})\,] = \Pr[\,\Pi_T(s,r) = (s_{i_1}, \ldots, s_{i_{|T|}})\,],$$

where the first probability is over the randomness of the simulator SIM *and the second probability is over the choice of r from R at random according to μ.*

A scheme is linear if the mapping that the dealer uses to generate the shares that are given to the parties is linear, as we formalize at the following definition.

Definition 2.3 (Linear Secret-Sharing Schemes). *Let $\Sigma = \langle \Pi, \mu \rangle$ be a secret-sharing scheme with domain of secrets S, where μ is a probability distribution on a set R and Π is a mapping from $S \times R$ to $S_1 \times S_2 \times \cdots \times S_n$. We say that Σ is a* linear secret-sharing scheme over a finite field $\mathbb{F}$ *if $S = \mathbb{F}$, the sets $R, S_1, \ldots, S_n$ are vector spaces over $\mathbb{F}$, Π is an $\mathbb{F}$-linear mapping, and μ is the uniform probability distribution over R.*

Next, we present the definition of k-uniform access structures. In such access structures, the authorized sets are all the subsets of parties of size greater than k and some of the subsets of parties of size k.

Definition 2.4 (Uniform Access Structures). *Let $P = \{P_1, \ldots, P_n\}$ be a set of parties. An access structure $\Gamma \subseteq 2^P$ is a k-uniform access structure, where $1 \leq k \leq n$, if all sets of size less than k are unauthorized, all sets of size greater than k are authorized, and each set of size exactly k can be either authorized or unauthorized.*

Now, we define threshold secret-sharing schemes, and give the known result for such schemes, as presented in [39].

Definition 2.5 (Threshold Secret-Sharing Schemes). *Let Σ be a secret-sharing scheme on a set of n parties P. We say that Σ is a* t-out-of-n secret-sharing scheme *if it realizes the access structure $\Gamma_{t,n} = \{A \subseteq P : |A| \geq t\}$.*

Claim 2.6 ([39]). *For every set of n parties P and for every $t \in [n]$, there is a linear t-out-of-n secret-sharing scheme realizing $\Gamma_{t,n} \subseteq 2^P$ for secrets of size ℓ in which the share size of every party is $\max\{\ell, \log n\}$.*

Fact 2.7 ([16]). *Let $\Gamma_1, \ldots, \Gamma_t$ be access structures over the same set of n parties, and let $\Gamma = \Gamma_1 \cup \cdots \cup \Gamma_t$ and $\Gamma' = \Gamma_1 \cap \cdots \cap \Gamma_t$. If there exist secret-sharing schemes with share size at most k realizing $\Gamma_1, \ldots, \Gamma_t$, then there exist secret-sharing schemes realizing Γ and Γ' with share size at most kt. If the former schemes are linear over a finite field $\mathbb{F}$, then there exist linear secret-sharing schemes over $\mathbb{F}$ realizing Γ and Γ' with share size at most kt.*

Conditional Disclosure of Secrets Protocols. Next we define k-party conditional disclosure of secrets (CDS) protocols, first presented in [28]. We consider a model where a set of k parties $P = \{P_1, \ldots, P_k\}$ hold a secret s and a common random string r. In addition, every party P_i holds an input x_i for some k-input function f. In a CDS protocol for f, for every $i \in [k]$, party P_i sends a message to a referee, based on r, s and x_i, such that the referee can reconstruct the secret s if $f(x_1, \ldots, x_k) = 1$, and it cannot learn any information about the secret if $f(x_1, \ldots, x_k) = 0$. Formally,

Definition 2.8 (Conditional Disclosure of Secrets Protocols – Syntax and Correctness). *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be some k-input function. A* k-party CDS protocol *$\mathcal{P}$ for f with domain of secrets S consists of:*

- *A finite domain of common random strings R, and k finite message domains $M_1, \ldots, M_k$.*
- *Deterministic message computation functions $\text{ENC}_1, \ldots, \text{ENC}_k$, where*

$$\text{ENC}_i : X_i \times S \times R \to M_i$$

  for every $i \in [k]$.
- *A deterministic reconstruction function*

$$\text{DEC} : X_1 \times \cdots \times X_k \times M_1 \times \cdots \times M_k \to \{0,1\}\,.$$

We say that a CDS protocol $\mathcal{P}$ is correct (with respect to f) if for every inputs $(x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 1$, every secret $s \in S$, and every common random string $r \in R$,

$$\text{DEC}(x_1, \ldots, x_k, \text{ENC}_1(x_1, s, r), \ldots, \text{ENC}_k(x_k, s, r)) = s.$$

The message size of a CDS protocol $\mathcal{P}$ is defined as the size of largest message sent by the parties, i.e., $\max_{1 \leq i \leq k}\{\log |M_i|\}$.

We define the privacy of CDS protocols with a simulator, i.e., given $x_1, \ldots, x_k$ such that $f(x_1, \ldots, x_k) = 0$, we can simulate the messages sent by the parties by a simulator that has access to $x_1, \ldots, x_k$ and does not know the secret, in such a way that one cannot distinguish between the messages sent by the parties and the messages generated by the simulator. That is, a CDS protocol is private if everything that can be learned from it can be learned from $x_1, \ldots, x_k$ without knowing the secret.

Definition 2.9 (Conditional Disclosure of Secrets Protocols – Privacy). *We say that a CDS protocol $\mathcal{P}$ is private (with respect to f) if there exists a randomized function* Sim*, called the* simulator*, such that for every inputs $(x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 0$, every secret $s \in S$, and every k messages $(m_1, \ldots, m_k) \in M_1 \times \cdots \times M_k$,*

$$\begin{aligned}&\Pr[\,\textsc{Sim}(x_1, \ldots, x_k) = (m_1, \ldots, m_k)\,] \\ &\quad = \Pr[\,(\textsc{Enc}_1(x_1, s, r), \ldots, \textsc{Enc}_k(x_k, s, r)) = (m_1, \ldots, m_k)\,],\end{aligned}$$

where the first probability is over the randomness of the simulator Sim *and the second probability is over the choice of r from R with uniform distribution (the same r is chosen for all encryptions).*

Private Simultaneous Messages Protocols. We next define k-party ad-hoc private simultaneous messages (PSM) protocols, as presented in [14]. Let $P = \{P_1, \ldots, P_n\}$ be a set of n parties. For every $i \in [n]$, party P_i holds an input x_i for some k-input function f, and a random string r_i. In an ad-hoc PSM protocol for f, only a subset of the parties $A \subseteq P$ participates, where each of them sends a single message to a referee, which is based on its input x_i and the random string r_i. If exactly k parties participate and send messages, that is, $A = \{P_{i_1}, \ldots, P_{i_k}\}$, where $i_1 < \cdots < i_k$, then the referee should be able to compute $f(x_{i_1}, \ldots, x_{i_k})$ using the k messages it gets, but should not learn any additional information about the inputs $x_{i_1}, \ldots, x_{i_k}$. The subset of participating parties A is selected in an ad-hoc manner, and, in particular, the participating parties are not aware of each other. The referee itself learns the set of parties A (since it receives messages directly from the parties in A). Below, we present the formal definition of ad-hoc PSM protocols.

Definition 2.10 (Ad-hoc Private Simultaneous Messages Protocols – Syntax and Correctness). *Let $P = \{P_1, \ldots, P_n\}$ be a set of parties and let $f : X^k \to Y$ be some k-input function. A* k-out-of-n ad-hoc PSM protocol *$\mathcal{P}$ for f consists of:*

- *A finite domain of common random strings R, and a finite message domain M.*
- *Deterministic message computation functions $\textsc{Enc}_1, \ldots, \textsc{Enc}_n$, where*

$$\textsc{Enc}_i : X \times R \to M$$

for every $i \in [n]$.

– *A deterministic reconstruction function*

$$\textsc{Dec} : \binom{P}{k} \times M^k \to Y.$$

We say that an ad-hoc PSM protocol $\mathcal{P}$ is correct (with respect to f) if for any set $A = \{P_{i_1}, \ldots, P_{i_k}\} \in \binom{P}{k}$, every inputs $(x_{i_1}, \ldots, x_{i_k}) \in X^k$, and every common random string $r \in R$,

$$\textsc{Dec}(A, \textsc{Enc}_{i_1}(x_{i_1}, r), \ldots, \textsc{Enc}_{i_k}(x_{i_k}, r)) = f(x_{i_1}, \ldots, x_{i_k}).$$

The message size of an ad-hoc PSM protocol $\mathcal{P}$ is the size of the messages sent by each of the parties, i.e., $\log |M|$.

Definition 2.11 (Ad-hoc Private Simultaneous Messages Protocols – Privacy). *We say that an ad-hoc PSM protocol $\mathcal{P}$ is private (with respect to f) if:*

– *There exists a randomized function* Sim*, called a* simulator*, such that for every $A = \{P_{i_1}, \ldots, P_{i_k}\} \in \binom{P}{k}$, every inputs $(x_{i_1}, \ldots, x_{i_k}) \in X^k$, and every k messages $(m_{i_1}, \ldots, m_{i_k}) \in M^k$,*

$$\begin{aligned} &\Pr[\, \textsc{Sim}(A, f(x_{i_1}, \ldots, x_{i_k})) = (m_{i_1}, \ldots, m_{i_k}) \,] \\ &= \Pr[\, (\textsc{Enc}_{i_1}(x_{i_1}, r), \ldots, \textsc{Enc}_{i_k}(x_{i_k}, r)) = (m_{i_1}, \ldots, m_{i_k}) \,], \end{aligned}$$

where the first probability is over the randomness of the simulator Sim *and the second probability is over the choice of r from R with uniform distribution (the same r is chosen for all encryptions).*

– *There exists a randomized function* Sim$'$*, called a* simulator*, such that for every $k' < k$, every $A' = \{P_{i_1}, \ldots, P_{i_{k'}}\} \in \binom{P}{k'}$, every inputs $(x_{i_1}, \ldots, x_{i_{k'}}) \in X^{k'}$, and every k' messages $(m_{i_1}, \ldots, m_{i_{k'}}) \in M^{k'}$,*

$$\begin{aligned} &\Pr[\, \textsc{Sim}'(A') = (m_{i_1}, \ldots, m_{i_{k'}}) \,] \\ &= \Pr[\, (\textsc{Enc}_{i_1}(x_{i_1}, r), \ldots, \textsc{Enc}_{i_{k'}}(x_{i_{k'}}, r)) = (m_{i_1}, \ldots, m_{i_{k'}}) \,], \end{aligned}$$

where the first probability is over the randomness of the simulator Sim$'$ *and the second probability is over the choice of r from R with uniform distribution (the same r is chosen for all encryptions).*

A PSM protocol is a k-out-k ad-hoc PSM protocol, where the privacy requirement only holds for sets of size k (we do not require that a referee that gets messages from less than k parties will not learn any information).

Notation. We denote the logarithmic function with base 2 and base e by log and ln, respectively. Additionally, we use the notation $[n]$ to denote the set $\{1, \ldots, n\}$. For $0 \leq \alpha \leq 1$, we denote the binary entropy of α by

$$h(\alpha) \stackrel{\text{def}}{=} -\alpha \log \alpha - (1-\alpha) \log(1-\alpha).$$

Next, we present an approximation of the binomial coefficients.

Fact 2.12. *For every k and every n such that $k \in [n]$, it holds that*

$$\binom{n}{k} = \Theta(k^{-1/2} \cdot 2^{h(k/n)n}).$$

3 Secret-Sharing Schemes Realizing General Access Structures from CDS Protocols

In this section we present a construction of secret-sharing schemes for a general access structure. The starting point of our results is a work by Liu and Vaikuntanathan [34], in which they presented the first general construction with share size $O(2^{cn})$ with a constant c smaller than 1. In the first part of the section, we give an outline of the construction in [34], presenting their results in terms of access structures. Our main result, is the following theorem.

Theorem 3.1. *Every access structure over n parties can be realized by a secret-sharing scheme with a total share size of $2^{0.892n+o(n)}$ and by a linear secret-sharing scheme with a total share size of $2^{0.942n+o(n)}$.*

We say that an access structure Γ can be *realized with an exponent of S* (resp., *linearly realized with an exponent of S*) if Γ can be realized by a secret-sharing scheme (resp., linear secret-sharing scheme) with shares of size at most $2^{Sn+o(n)}$ where n is the number of participants.[1]

3.1 Our construction

Following Liu and Vaikuntanathan [34], we decompose an access structure Γ to three parts: a bottom part (that handles small sets), middle part (that handles medium-size sets) and a top part (that handles large sets). Formally, we have the following proposition.

Proposition 3.2 ([34]). *For every access structure Γ over a set of n participants, and every slice parameter $\delta \in (0, \frac{1}{2})$, define the following access structures over the same set of participants.*

$$\Gamma_{\text{bot}} : A \in \Gamma_{\text{bot}} \quad \textit{iff} \quad \exists A' \in \Gamma \textit{ s.t. } A' \subseteq A \textit{ and } |A'| \leq (\frac{1}{2} - \delta)n,$$

$$\Gamma_{\text{mid}} : A \in \Gamma_{\text{mid}} \quad \textit{iff} \quad A \in \Gamma \textit{ and } (\frac{1}{2} - \delta)n \leq |A| \leq (\frac{1}{2} + \delta)n, \textit{ or } |A| \geq (\frac{1}{2} + \delta)n$$

$$\Gamma_{\text{top}} : A \notin \Gamma_{\text{top}} \quad \textit{iff} \quad \exists A' \notin \Gamma \textit{ s.t. } A \subseteq A' \textit{ and } |A'| \geq (\frac{1}{2} + \delta)n.$$

Then $\Gamma = \Gamma_{\text{top}} \cap (\Gamma_{\text{mid}} \cup \Gamma_{\text{bot}})$. Consequently, if $\Gamma_{\text{top}}, \Gamma_{\text{mid}}$, and Γ_{bot} can be realized (resp., linearly realized) with exponent of S then so is Γ.

[1] Formally, such a statement implicitly refers to an infinite sequence of (collections of) access structures that is parameterized by the number of participants n.

The "consequently" part follows from standard closure properties of secret-sharing schemes (see Fact 2.7). Thus realizing Γ reduces to realizing Γ_{top}, Γ_{bot}, and Γ_{mid}. The main work in [34] is devoted to realizing the access structure Γ_{mid}. Their main construction can be summarized as follows.

Lemma 3.3 ([34]). *For every access structure Γ and every slice parameter $\delta \in (0, \frac{1}{2})$, the access structure Γ_{mid} can be realized with an exponent of*

$$\mathbf{M}(\delta) = h(0.5 - \delta) + 0.2h(10\delta) + 10\delta - 0.2\log(10),$$

and can be linearly realized with an exponent of

$$\mathbf{M}_\ell(\delta) = h(0.5 - \delta) + 0.2h(10\delta) + 10\delta - 0.1\log(10).^2$$

The extreme slices. Liu and Vaikuntanathan [34] realized Γ_{top} and Γ_{bot} with an exponent of $h(\frac{1}{2} + \delta)$ by exploiting the fact that the number of authorized (or non-authorized) sets is exponential in $h(\frac{1}{2} + \delta)$. (The actual implementation is based on the classical schemes of [32].) We show that the nice structure of these access structures can be further exploited.

In particular, for a *covering parameter* α, the minimal authorized sets of Γ_{bot} can be covered by exponentially-many αn-subsets of n. (A dual statement applies to the maximal unauthorized sets of Γ_{top}.) This property allows us to realize Γ_{bot} and Γ_{top} by decomposing each of them into (exponentially) many access structures over αn parties and realizing each access structure via a general secret-sharing scheme. Overall, we get a tradeoff between the size of the decomposition (i.e., number of sub-access structures) and the number of players αn in each part. Formally, in Section 3.2 we prove the following statement.

Lemma 3.4. *Suppose that every access structure can be realized (resp., linearly realized) with an exponent of S. Then, for every covering parameter $\alpha \in (\frac{1}{2}, 1)$, every access structure Γ and every slice parameter $\delta \in (0, \frac{1}{2})$, the access structures Γ_{top} and Γ_{bot} can be realized (resp., linearly realized) with an exponent of*

$$\mathbf{X}(S, \delta, \alpha) \stackrel{def}{=} \alpha S + h(0.5 - \delta) - h\left((0.5 - \delta)/\alpha\right)\alpha.^3$$

By combining Lemmas 3.3 and 3.4 with Proposition 3.2, we derive the following Theorem.

Theorem 3.5. *Suppose that every access structure can be realized (resp., linearly realized) with an exponent of S (resp., S_ℓ). Then, for every covering parameter $\alpha \in (\frac{1}{2}, 1)$ and slice parameter $\delta \in (0, \frac{1}{2})$, every access structure can be realized with an exponent of* $\max\left(\mathbf{M}(\delta), \mathbf{X}(S, \delta, \alpha)\right)$, *and can be linearly realized with an exponent of* $\max\left(\mathbf{M}_\ell(\delta), \mathbf{X}(S_\ell, \delta, \alpha)\right)$.

[2] The notation $\mathbf{M}$ stands for "middle".
[3] The notation $\mathbf{X}$ stands for eXternal slices.

We can improve the secret sharing exponent by applying Theorem 3.5 recursively as follows. Start with the Liu-Vaikuntanathan bound $\mathbf{S}_{\mathrm{LV}} = 0.994$ as an initial value, and iterate with carefully chosen values for δ and α.

Example 3.6. Consider a single application of Theorem 3.5 starting with $\mathbf{S}_{\mathrm{LV}} = 0.994$ and taking $\delta = 0.037$ and $\alpha = 0.99$. In this case, $\mathbf{M}(\delta) < 0.897$ and $\mathbf{X}(\mathbf{S}_{\mathrm{LV}}, \delta, \alpha) < 0.9931$, thus we get an exponent smaller than 0.9931.

Since each step of the recursion is parameterized by both δ and α, the problem of finding the best choice of parameters in every step of the recursion becomes a non-trivial optimization problem. In Section 3.3, we analyze the recursive process and derive an analytic expression for the infimum of the process (over all sequences of (δ_i, α_i)). This leads to a general scheme with an exponent of 0.897 and a linear scheme with an exponent of 0.955. Finally, an additional (minor) improvement is obtained by analyzing a low-level optimization to the middle slice that was suggested by [34] (see Section 3.4). This leads to Theorem 3.1.

3.2 Realizing $\Gamma_{\mathbf{bot}}$ and $\Gamma_{\mathbf{top}}$ (Proof of Lemma 3.4)

We start by introducing a fact about *combinatorial covering designs* by Erdös and Spenser:

Fact 3.7 ([25]). *Let P be a set of size n. For every positive integers $c \leq a \leq n$, there exists a family $\mathcal{G} = \{G_i\}_{i=1}^{L}$ of a-subsets of P, such that every c-subset of P is contained in at least one member of $\mathcal{G}$, and $L = L(n,a,c) = O(\binom{n}{c}\log\binom{a}{c})/\binom{a}{c})$.*

We next prove Lemma 3.4.

Proof (of Lemma 3.4). Let $a = \alpha n$ and $c = (0.5-\delta)n$ and let $\mathcal{G} = \{G_i\}_{i\in[L]}$ be the family of a-subsets of $P = \{P_1, \ldots, P_n\}$ promised by Fact 3.7. Using Fact 2.12, the number of sets L satisfies

$$\log L \leq n(h(0.5-\delta) - h((0.5-\delta)/\alpha)\alpha + o(1)).$$

Hence, to prove the lemma it suffices to realize Γ_{bot} and Γ_{top} with share size of $L \cdot 2^{S\alpha n + o(n)}$. Towards this end, we decompose Γ_{bot} and Γ_{top} according to $\mathcal{G}$ as follows.

Γ_{bot}: Let $\mathcal{T}$ be the set of minimal authorized sets of Γ_{bot}. Recall that all these sets are of size of at most c. For every $i \in [L]$, let $\mathcal{T}_i = \{T \in \mathcal{T} : T \subseteq G_i\}$, and let Γ_i be the access structure whose minimal authorized sets are the sets in $\mathcal{T}_i$. By Fact 3.7, $\mathcal{T} = \bigcup \mathcal{T}_i$ and therefore $\Gamma_{\mathrm{bot}} = \bigcup_{i\in[L]} \Gamma_i$. Indeed, both in the RHS and in the LHS, A is an authorized set iff there exists some T in $\mathcal{T} = \bigcup \mathcal{T}_i$ such that $T \subseteq A$. We further note that every minimal authorized set in Γ_i is a subset of G_i and therefore Γ_i can be implemented as an access structure over αn parties with share size of $2^{S\alpha n + o(n)}$. To share a secret s according to $\Gamma_{\mathrm{bot}} = \bigcup_{i\in[L]} \Gamma_i$, for every $i \in [L]$ independently share s via the scheme of Γ_i. The share size of the resulting scheme realizing Γ_{bot} is $L \cdot 2^{S\alpha n + o(n)}$, as required.

Γ_{top}*:* We use a dual construction for Γ_{top}. Let $\mathcal{T}'$ be the set of maximal unauthorized sets of Γ_{top}. Recall that all these sets are of size at least $n - c$. For every $i \in [L]$, let $\mathcal{T}'_i = \{T \in \mathcal{T}' : \overline{G_i} \subseteq T\} = \{T \in \mathcal{T}' : \overline{T} \subseteq G_i\}$ and let Γ'_i be the access structure whose maximal unauthorized sets are the sets in $\mathcal{T}_i$. By Fact 3.7, $\mathcal{T}' = \bigcup \mathcal{T}'_i$ and therefore $\Gamma_{\text{top}} = \bigcap_{i\in[L]} \Gamma'_i$. Indeed, both in the RHS and in the LHS, A is an unauthorized set iff $A \subseteq T$ for some T in $\mathcal{T}' = \bigcup \mathcal{T}'_i$. We further note that all minimal authorized sets of Γ'_i are subsets of $\bigcap_{T\in\mathcal{T}'_i} \overline{T} \subseteq G_i$, and therefore Γ_i can be implemented as an access structure over αn parties with share size of $2^{S\alpha n+o(n)}$. To share a secret s according to Γ_{top}, sample L random elements $s_1, \ldots, s_L$ in the domain of s satisfying $s = s_1 + \ldots + s_L$, and share s_i via the scheme for Γ_i. A set A can reconstruct the secret iff it can reconstruct each s_i iff $A \in \Gamma'_i$ for every i iff $A \in \bigcap_{i\in[L]} \Gamma'_i = \Gamma_{\text{top}}$. Thus, we can realize Γ_{top} with share size of

$$L \cdot 2^{(\alpha S+o(1))n} = 2^{(\alpha S+h(0.5-\delta)-h((0.5-\delta)/\alpha)\alpha+o(1))n},$$

as required. □

3.3 Analyzing the Recursion

In this section, we analyze the exponent achievable by repeated applications of Theorem 3.5 by considering the following single-player game.

The exponent game. The goal of the player is to minimize a positive number S. The value of S is initialized to the LV-exponent 0.994, and can be updated by making an arbitrary number of moves. In each move the player can choose $\delta \in (0, \frac{1}{2})$ and $\alpha \in (\frac{1}{2}, 1)$, if $S < \max(\mathbf{X}(S, \delta, \alpha), \mathbf{M}(\delta))$, update S to $\max(\mathbf{X}(S, \delta, \alpha), \mathbf{M}(\delta))$; otherwise, S remains unchanged.

Recall that the function $\mathbf{X}(S, \delta, \alpha)$ represents the exponent of the external slices and the function $\mathbf{M}(\delta)$ represents the exponent of middle slice. We denote by **opt** the infimum of S over all finite sequences of (δ_i, α_i). Our goal is to determine **opt**. A (δ, α)-move improves S if and only if (1) $\mathbf{X}(S, \delta, \alpha) < S$ and (2) $\mathbf{M}(\delta) < S$. If the first condition holds we say that S is X-improved by (δ, α). We begin by showing that the question of whether a given S can be X-improved by a (δ, α)-move depends only on δ and S (and is independent of α and n).

Lemma 3.8. *Fix a parameter* $\delta \in (0, \frac{1}{2})$ *and let* $\mathbf{X}'(\delta) \stackrel{def}{=} h(0.5 - \delta) - (0.5 - \delta) \cdot \log\left((0.5 + \delta)/(0.5 - \delta)\right)$.

- *If* $S \leq \mathbf{X}'(\delta)$*, then there does not exist any* α *for which* $S > \mathbf{X}(S, \delta, \alpha)$.
- *For every* $S' > \mathbf{X}'(\delta)$ *there exists an* $\alpha < 1$ *such that* $\mathbf{X}(S, \delta, \alpha) \leq \alpha S + (1 - \alpha)S'$ *for every* $S > \mathbf{X}'(\delta)$.

Proof. Fix some S. The exponent S is X-improved by (δ, α) if and only if

$$\frac{h(0.5 - \delta) - h\left(\frac{0.5-\delta}{\alpha}\right)\alpha}{1 - \alpha} < S. \tag{1}$$

Denote the LHS by $\mathbf{X}'(\delta,\alpha)$. Clearly, S can be X-improved by (δ,α) if and only if S is larger than $\inf_\alpha(\mathbf{X}'(\delta,\alpha))$ (assuming that the infimum exists). We next show that $\inf_\alpha(\mathbf{X}'(\delta,\alpha)) = \mathbf{X}'(\delta)$. Indeed, for any fixed δ, the function $\mathbf{X}'(\delta,\alpha)$ is monotonically decreasing with α, and since $\alpha < 1$, we get that

$$\inf_\alpha(\mathbf{X}'(\delta,\alpha)) = \lim_{\alpha\to 1} \frac{h(0.5-\delta) - h\left((0.5-\delta)/\alpha\right)\alpha}{1-\alpha},$$

which by l'Hôpitals Rule, simplifies to $\mathbf{X}'(\delta)$. The first item of the claim follows.

For the second item, take any $\alpha \in (0,1)$ such that

$$\frac{h(0.5-\delta) - h\left(\frac{0.5-\delta}{\alpha}\right)\alpha}{1-\alpha} \le S'$$

(by the definition of the limit and since $S' > \mathbf{X}'(\delta)$, such α exists). Thus,

$$\mathbf{X}(S,\delta,\alpha) = \alpha S + h(0.5-\delta) - h\left((0.5-\delta)/\alpha\right)\alpha \le \alpha S + (1-\alpha)S'.$$

Note that the choice of α is independent of S (as long as $S > \mathbf{X}'(\delta)$). □

Lemma 3.8 takes into account only the effect of the outer slices, Γ_{top} and Γ_{bot}. Recall, however, that the cost of the medium slice Γ_{mid} prevents us from going below $\mathbf{M}(\delta)$. Let $\delta^\star \in (0, 0.5)$ denote the positive value that satisfies $\mathbf{X}'(\delta^\star) = \mathbf{M}(\delta^\star)$. Let us denote by $S^\star$ the value of $\mathbf{X}'(\delta^\star) = \mathbf{M}(\delta^\star)$. The curves of $\mathbf{M}(\delta)$ and $\mathbf{X}(\delta)$ are depicted in Fig. 1, and $\delta^\star \approx 0.037, S^\star \approx 0.897$. The following two claims show that the infimum of the game, **opt**, equals to $S^\star$.

Claim 3.9. *For every constant $S'' > S^\star$ there exists an $\alpha < 1$ and an integer i (where α and i are independent of n) such that a sequence of i (δ,α)-moves improve the exponent to S''.*

Proof. Choose any constant S' such that $S^\star < S' < S''$ and let α be a constant guaranteed by Lemma 3.8 for $\delta^\star$ and S', that is for every $S > S'$, the exponent S can be improved to $\alpha S + (1-\alpha)S'$. Furthermore, let $S_0 = 0.994$ be the exponent of the secret-sharing scheme of [34] and define $S_j = \alpha S_{j-1} + (1-\alpha)S'$ for every $j > 0$. By Lemma 3.8, the exponent S_j can be achieved after j (δ,α)-moves. By induction, $S_j = \alpha^j S_0 + (1-\alpha^j)S' < \alpha^j S_0 + S'$. Taking an integer i such that $\alpha^i \le (S'' - S')/S_0$ completes the proof. □

Claim 3.10. *There is no (δ,α)-move that takes a value $A > S^\star$ to a value $B < S^\star$. Consequently, any finite number of steps ends in a value $S > S^\star$ and* **opt** $\ge S^\star$.

Proof. Assume towards a contradiction that there is some (δ,α)-move that takes some value $A > S^\star$ to some value $B < S^\star$. If $\delta > \delta^\star$, then $B \ge \mathbf{M}(\delta) > S^\star$, contradiction. Thus, we can assume that $\delta \le \delta^\star$. Choose some $S \in (B, S^\star)$. Observe that a (δ,α)-move from S leads to a value $D \le B < S$, and so this move improves S. It follows that $S > \mathbf{X}(S,\delta,\alpha)$ and, by Lemma 3.8, $S > \mathbf{X}'(\delta) \ge \mathbf{X}'(\delta^\star) = S^\star$, contradiction.

We conclude that in both cases $\delta > \delta^\star$ and $\delta \le \delta^\star$, we get a contradiction, and $B \ge S^\star$. □

Overall we get that $\mathbf{opt} = S^\star$ which is about 0.897.

Theorem 3.11. *Every access structure can be realized with share size* $2^{(0.897+o(1))n}$.

Remark 3.12. We note that our analysis holds even if the function $\mathbf{M}(\delta)$ is replaced with a different function that represents the exponent of the middle slice. That is, for any choice of $\mathbf{M}(\delta)$ the value of $\mathbf{opt}$ equals to $\inf_\delta \max(\mathbf{X}'(\delta), \mathbf{M}(\delta))$ (assuming that the initial starting point is over the $\mathbf{M}(\delta)$ curve).

In particular, the following theorem is obtained by replacing $\mathbf{M}(\delta)$ with the exponent $\mathbf{M}_\ell(\delta)$ for a linear realization of the middle layer (from Lemma 3.3).

Theorem 3.13. *Every access structure can be linearly realized with share size of* $2^{(0.955+o(1))n}$.

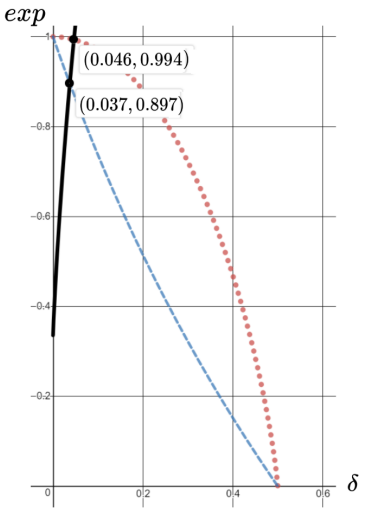

Fig. 1. A description of the functions $\mathbf{M}(\delta)$ and $\mathbf{X}'(\delta)$. The horizontal axis represents the value of δ and the vertical axis represents the resulting exponents. The solid black curve corresponds to the exponent $\mathbf{M}(\delta)$ of the middle slice Γ_{mid}, as defined in Lemma 3.3 (the minor improvements of Section 3.4 do not appear here). The function $\mathbf{X}'(\delta)$ appears as the dashed blue line. For comparison, we plot in the dotted red line the exponent that is achieved for Γ_{top} and Γ_{bot} via the simple (non-recursive) construction from [34]. Our exponent appears as the y-coordinate of the intersection of the black and blue curves, and the exponent of [34] appears at the y-coordinate of the intersection of the red and black curves.

3.4 Minor improvement of share size for Γ_{mid}

In this section we give a tighter analysis for the constructions of $\mathbf{M}(\delta)$ and $\mathbf{M}_\ell(\delta)$ from [34]. These ideas were suggested in [34], but were not implemented.

Lemma 3.14. *For every access structure Γ and every slice parameter $\delta \in (0, \frac{1}{2})$, the access structure Γ_{mid} can be realized with an exponent of*

$$\mathbf{M}(\delta) = h(0.5-\delta) + 0.2h(10\delta) + 2\log(26)\delta - 0.2\log(10),$$

and can be linearly realized with an exponent of

$$\mathbf{M}_\ell(\delta) = h(0.5-\delta) + 0.2h(10\delta) + 2\log(26)\delta - 0.1\log(10).$$

The above expressions slightly improves over the ones obtained in Lemma 3.3. In particular, the third summand in both $\mathbf{M}(\delta)$ and $\mathbf{M}_\ell(\delta)$ is reduced from 10δ to $2\log(26)\delta$.

Proof. We assume familiarity with the construction of [34]. In the original analysis of reduction 4 in [34, Section 3.4], the expression 10δ is added to the exponent (of both $\mathbf{M}(\delta)$ and $\mathbf{M}_\ell(\delta)$) due to an enumeration over all possible subsets that are taken from a universe of size $10\delta n$. It is noted there that it actually suffices to enumerate only over subsets T that satisfy the following condition. For a given (fixed) partition of the universe to $2\delta n$ bins of size 5 each, the set T must contain at least 2 elements from each bin. The number of such sets is $(2^5 - \binom{5}{0} - \binom{5}{1})^{2\delta n} = 2^{2\log(26)\delta n}$, and so the lemma follows. □

We can further improve the exponent of the linear scheme by reducing the last summand as follows.

Lemma 3.15. *For every access structure Γ and every slice parameter $\delta \in (0, \frac{1}{2})$, the access structure Γ_{mid} can be linearly realized with an exponent of $\mathbf{M}_\ell(\delta) = h(0.5-\delta) + 0.2h(10\delta) + 2\log(26)\delta - (0.1+\delta)\log(10)$.*

Proof. Again, we assume familiarity with the construction of [34]. The last reduction of [34, Section 3.5] utilizes a protocol for *conditional disclosure of secrets* (CDS) with an input size of $\binom{n/k}{a/k}^k$ for $a = \frac{1}{2} - \delta$ and $k = n/5$. As the authors note, for this choice of parameters, the input size of the CDS is actually $\binom{n/k}{a/k}^{k-2\delta}$. (In general, this holds whenever $\binom{n/k}{\lfloor a/k \rfloor} = \binom{n/k}{\lceil a/k \rceil}$.) This improvement becomes useful in the linear case (which employs linear CDS), and eventually it leads to the improvement stated in the lemma. □

Proof (proof of Theorem 3.1). As stated in Remark 3.12, the analysis of the recursive process holds when $\mathbf{M}(\delta)$ is updated to some $\mathbf{M}'(\delta)$, and the new infimum of the exponent game becomes $\mathbf{X}'(\delta^\star)$, where $\delta^\star$ satisfies $\mathbf{X}'(\delta^\star) = \mathbf{M}'(\delta^\star)$. By using the bounds obtained in Lemmas 3.14 and 3.15, we derive Theorem 3.1. □

4 Secret-Sharing Schemes Realizing k-Uniform Access Structures with Long Secrets

In this section, we present a construction of secret-sharing schemes for k-uniform access structures on n parties using k^2-party CDS protocols. Using the CDS protocols of [2] with long secrets, we obtain secret-sharing schemes for long secrets in which the share size of every party is only $O(k^2)$ times the secret size.

In [2], it was shown how to construct a secret-sharing scheme realizing any k-uniform access structure Γ in which the share size of every party is $O(e^k)$ times the message size, with big secrets. To construct this scheme, they used a family of perfect hash functions from $[n]$ to $[k]$, where each such function defines a k-uniform access structure, and use a k-party CDS protocol to realize every such access structure. Since the number of perfect hash functions for sets of size k and range of size k is bigger than e^k, each share in the secret-sharing scheme of [2] contains $O(e^k)$ messages of the CDS protocol. We improve their construction by taking a family of perfect hash functions from $[n]$ to $[k^2]$, and construct a secret-sharing scheme using a k^2-party CDS protocol for every function in this family, such that the resulting scheme realizes Γ.

The definition of a family of perfect hash functions is presented next.

Definition 4.1 (Families of Perfect Hash Functions). *A set of functions $H_{n,k,t} = \{h_i : [n] \to [t] : i \in [\ell]\}$ is a family of perfect hash functions if for every set $A \subseteq [n]$ such that $|A| = k$ there exists at least one index $i \in [\ell]$ such that $|h_i(A)| = |\{h_i(a) : a \in A\}| = k$, i.e., h_i restricted to A is one-to-one.*

To construct a secret-sharing scheme that realizes the k-uniform access structure Γ, we construct a scheme, using a CDS protocol for the function f (defined in Definition 4.2), that realizes the access structure Γ_h (defined in Definition 4.3), for every function h among a family of perfect hash functions.

Definition 4.2 (The Function f). *Let Γ be a k-uniform access structure with n parties. The k^2-input function $f : \{0, 1, \ldots, n\}^{k^2} \to \{0, 1\}$ is the function that satisfies $f(x_1, \ldots, x_{k^2}) = 1$ if and only if $\{P_{x_i} : i \in [k^2], x_i \neq 0\} \in \Gamma$.*

For example, if $k = 2$, $n = 5$, and the authorized sets of size $k = 2$ are exactly $\{P_1, P_2\}, \{P_3, P_5\}$, then $f(1, 3, 5, 0) = f(1, 2, 0, 0) = f(0, 2, 0, 1) = f(3, 0, 0, 5) = 1$ and $f(0, 0, 0, 0) = f(0, 2, 0, 0) = f(0, 2, 3, 0) = f(2, 0, 0, 5) = 0$.

Definition 4.3 (The Access Structure Γ_h). *Let Γ be a k-uniform access structure with n parties and let $h : [n] \to [k^2]$ be a function. The k-uniform access structure Γ_h is the access structure that contains all the subsets of parties of size greater than k, and all authorized subsets from Γ of size k such that h restricted to the indices of the parties of such subset is one-to-one. That is,*

$$\Gamma_h = \{A \subseteq P : |A| > k\} \cup \{A \subseteq P : A \in \Gamma, |A| = k, \text{ and } |\{h(j) : P_j \in A\}| = k\}.$$

Using a simple probabilistic proof, we show that there exists a family of perfect hash function $H_{n,k,k^2} = \{h_1, \ldots, h_\ell\}$ with $\ell = \Theta(k \cdot \log n)$ functions. Moreover, if H_{n,k,k^2} is a family of perfect hash functions, then $\Gamma = \cup_{h \in H_{n,k,k^2}} \Gamma_h$. Thus, constructing secret-sharing schemes realizing Γ_h for every $h \in H_{n,k,k^2}$, we get a secret-sharing scheme realizing Γ.

We start with a scheme that realizes the k-uniform access structure Γ_h, as defined in Definition 4.3; this scheme uses a CDS protocol for f, as defined in Definition 4.2. The scheme is described in Fig. 2; we next give an informal description. For every $j \in [n]$, we give the message of the $h(j)$th party in the CDS protocol when holding the input j, and for every $i \in [k^2]$, we share the message of the ith party in the CDS protocol when holding the input 0 using a k-out-of-k^2 scheme among the parties P_j for which $h(j) \neq i$.

Every authorized set $A \in \Gamma_h$ can reconstruct the secret, since every $P_j \in A$ gets the message of the $h(j)$th party in the CDS protocol when holding the input j, and the parties in A can reconstruct the messages of the other parties in the CDS protocol from the k-out-of-k^2 scheme, because $|\{h(j) : P_j \in A\}| = k$. Thus, the parties in A can reconstruct the secret since they hold messages for a 1-input of f.

Every unauthorized set $A \notin \Gamma$ that does not collide on h (that is, for every two different parties $P_j, P_{j'} \in A$ it must hold that $h(j) \neq h(j')$), the parties in A cannot learn any other messages except for the above mentioned messages. Thus, if $|A| = k$ then the parties in A hold messages for a 0-input of f, so by the privacy of the CDS protocol for f they cannot learn any information about the secret.

However, if A collides on h, then the parties in A hold two different messages of the same party in the CDS protocol for f, and CDS protocols cannot ensure any privacy in this scenario. To overcome this problem, we choose two random elements s_1, s_2 from the domain of secrets such that $s = s_1 + s_2$. We share s_1 using a k-out-of-k^2 scheme and give the $h(j)$th share to party P_j, and apply the above scheme using a CDS protocol for f with the secret s_2. Now, if A collides on h, the parties in A may learn information about s_2, but they cannot learn s_1, so the privacy of the scheme holds.

Lemma 4.4. *Let Γ be a k-uniform access structure with n parties, and let $h : [n] \to [k^2]$ be a function. Assume that for every k-input function $f : [N]^k \to \{0,1\}$ there is a k-party CDS protocol for f, for secrets of size t, in which the message size is $c(k, N, t)$. Then, the scheme Σ_h described in Fig. 2, is a secret-sharing scheme for secrets of size t realizing Γ_h in which the share size of every party is $O(\log n + k^2 \cdot c(k^2, n+1, t))$.*

Proof. Let $s \in S$ be the secret, where S is a finite set, and let $t = \log |S|$. Let $\mathcal{P}$ be a k^2-party CDS protocol for the function $f : \{0, 1, \ldots, n\}^{k^2} \to \{0,1\}$ defined in Definition 4.2 with message size $c(k^2, n+1, t)$.

We prove that the secret-sharing scheme Σ_h realizes the k-uniform access structure Γ_h with share size as in the lemma. First, we prove that in scheme Σ_h, every authorized set (i.e., a set in Γ_h) can reconstruct the secret s. Then,

The secret: An element $s \in S$.
The scheme: Let $f : \{0,1,\ldots,n\}^{k^2} \to \{0,1\}$ be a k^2-input function as in Definition 4.2, and let $\mathcal{P}$ be a k^2-party CDS protocol for f.

1. Share the secret s using a $(k+1)$-out-of-n secret-sharing scheme among the parties in P, that is, for every $j \in [n]$, give the jth share from this scheme to party P_j.
2. Choose a random element $s_1 \in S$, and let $s_2 = (s_1 + s) \mod |S|$.
3. Share the element s_1 using a k-out-of-k^2 secret-sharing scheme. For every $j \in [n]$, we give the $h(j)$th share from this scheme to party P_j.
4. Apply the k^2-party CDS protocol $\mathcal{P}$ to the k^2-input function f with the secret s_2 and a common random string r that is chosen at random. For every $i \in [k^2]$, let $m_{i,x}$ be the message of the ith party in the CDS protocol $\mathcal{P}$ when holding the input $x \in \{0,1,\ldots,n\}$.
5. For every $j \in [n]$, give the message $m_{h(j),j}$ to party P_j.
6. For every $i \in [k^2]$, share the message $m_{i,0}$ using a k-out-of-k^2 secret-sharing scheme. For every $j \in [n]$ such that $h(j) \neq i$, give the $h(j)$th share from this scheme to party P_j.

Fig. 2. A secret-sharing scheme Σ_h realizing the k-uniform access structure Γ_h.

we prove that every unauthorized set (i.e., a set not in Γ_h) cannot learn any information about s_1 or s_2, and, thus, it cannot learn any information about the secret s.

Correctness. Every authorized set of size greater than k can reconstruct the secret using the $(k+1)$-out-of-n secret-sharing scheme described in step 1 of the scheme Σ_h.

For an authorized set A of size k, it holds that $|\{h(j) : P_j \in A\}| = k$, i.e., the parties in A hold k different shares of s_1 from the k-out-of-k^2 scheme described in step 3 of the scheme Σ_h. Thus, the parties in A can reconstruct s_1 from this scheme.

Next, let $J = \{h(j) : P_j \in A\}$. Every party $P_j \in A$ holds the message $m_{h(j),j}$ of the CDS protocol $\mathcal{P}$ for f with the secret s_2, as described in step 5 of the scheme Σ_h. For every $i \in [k^2]$ such that $i \notin J$, the parties in A hold k different shares from the k-out-of-k^2 scheme in which the secret is the message $m_{i,0}$ of the CDS protocol $\mathcal{P}$ for f with the secret s_2, as described in step 6 of the scheme Σ_h. Hence, the parties in A can reconstruct $m_{i,0}$. Overall, the parties in A hold the messages $(m_{h(j),j})_{P_j \in A}$ and $(m_{i,0})_{i \notin J}$ of the CDS protocol $\mathcal{P}$ for f with the secret s_2, that is, the messages of the k^2 parties in the CDS protocol $\mathcal{P}$ for the secret s_2 when holding a 1-input of f. Thus, the parties in A can reconstruct s_2 and, together with s_1, they can reconstruct the secret s.

Privacy. Let A be an unauthorized set and let $J = \{h(j) : P_j \in A\}$. Observe that the size of A is at most k. If $|J| < k$, then the parties in A hold less than k shares from the k-out-of-k^2 scheme for s_1 in step 3, and so the parties in A cannot learn any information about the secret s.

If $|J| = k$, then $A \notin \Gamma$. For every $h(j) \in J$, the parties in A get only $k-1$ shares from the k-out-of-k^2 scheme for $m_{h(j),0}$ (as P_j does not get such share), so they cannot learn $m_{h(j),0}$. This implies that the parties in A hold only the messages $(m_{h(j),j})_{P_j \in A}$ and $(m_{i,0})_{i \notin J}$ of the CDS protocol $\mathcal{P}$ for f with the secret s_2; these messages are for a zero-input of f. By the privacy of the CDS protocol $\mathcal{P}$, the parties in A cannot learn any information about s_2, so they cannot reconstruct the secret s in the scheme Σ_h.

Formally, a simulator SIM for a set $A \notin \Gamma$ such that $|\{h(j) : P_j \in A\}| = k$ applies the simulator of the $(k+1)$-out-of-n secret-sharing scheme with the set A and returns the output of this simulator. Then, the simulator SIM chooses a random element $s'_1 \in S$ and shares s'_1 using a k-out-of-k^2 secret-sharing scheme, and for every j such that $P_j \in A$ it returns the $h(j)$th share from this scheme as part of the share of party P_j. Finally, SIM applies the simulator of the CDS protocol $\mathcal{P}$ with inputs $y_1, \ldots, y_{k^2}$, where $y_{h(j)} = j$ for every $P_j \in A$ and all other $k^2 - k$ y_i's are set to zero. It lets m'_i be the output of the simulator of $\mathcal{P}$ for the message of the ith party in $\mathcal{P}$. For every j such that $P_j \in A$, the simulator SIM returns $m'_{h(j)}$ as part of the share of party P_j. For every $i \notin \{h(j) : P_j \in A\}$, it shares m'_i using a k-out-of-k^2 secret-sharing scheme (this message is for an input $y_i = 0$) and for every $P_j \in A$ it gives the $h(j)$th share from this scheme as part of the share of party P_j. Finally, for every $P_j \in A$, it invokes the simulator of Shamir's scheme to generate $k-1$ shares for the parties $\left\{P_{h(j')} : P_{j'} \in A \setminus \{P_j\}\right\}$ and gives $P_{j'} \in A$ the share of $P_{h(j')}$.

SHARE SIZE. The share size of every party in the scheme Σ_h is $O(k^2 \cdot c(k^2, n+1, t) + \max\{t, \log n\}) = O(\log n + k^2 \cdot c(k^2, n+1, t))$. □

Using a simple probabilistic argument, we show the existence of a family of perfect hash functions with a small number of functions (satisfying a stronger requirement than in Definition 4.1).

Lemma 4.5. *There exists a family of perfect hash functions $H_{n,k,k^2} = \left\{h_i : [n] \to [k^2] : i \in [\ell]\right\}$, where $\ell = \Theta(k \cdot \log n)$, such that for every subset $A \subseteq [n]$ there are at least $\ell/4$ functions $h \in H_{n,k,k^2}$ for which $|h(A)| = k$.*

Proof. We show that there exists a family of hash functions H_{n,k,k^2} as above with $\Theta(k \cdot \log n)$ functions using the probabilistic method.

As a first step in the proof, we choose with uniform distribution a function $h : [n] \to [k^2]$, and fix a subset $A \subseteq [n]$ such that $|A| = k$. Then,

$$\begin{aligned}\Pr[\,|h(A)| \neq k] &= \Pr[\exists j_1, j_2 \in A, j_1 \neq j_2 : h(j_1) = h(j_2)\,] \\ &\leq \sum_{\substack{j_1, j_2 \in A \\ j_1 \neq j_2}} \Pr[\,h(j_1) = h(j_2)\,] = \binom{k}{2} \cdot \frac{1}{k^2} < \frac{1}{2}.\end{aligned}$$

Next, we show that if we choose at random $\ell = 16 \cdot k \cdot \ln n = \Theta(k \cdot \log n)$ functions as above, we can get the desired family $H_{n,k,k^2} = \{h_1, \ldots, h_\ell\}$.

We bound the probability that for a given subset $A \subseteq [n]$ of size k, there exist at most $\ell/4$ functions $h \in H_{n,k,k^2}$ that we choose at random, such that

$|h(A)| = k$. For every $i \in [\ell]$, let X_i be a boolean random variable such that $X_i = 1$ if $|h_i(A)| = k$ and $X_i = 0$ otherwise. Additionally, let $X = \sum_{i=1}^{\ell} X_i$, i.e., X is the number of hash functions h_i, for $i \in [\ell]$, such that $|h_i(A)| = k$. As we have shown above, $\Pr[\, X_i = 0\,] = \Pr[\, |h_i(A)| \neq k\,] < \frac{1}{2}$, so by linearity of expectation, $E(X) = \sum_{i=1}^{\ell} E(X_i) = \sum_{i=1}^{\ell} \Pr[\, X_i = 1\,] > \ell \cdot \frac{1}{2} = \frac{\ell}{2}$. Then, using the Chernoff bound that says that $\Pr[\, X \leq (1-\delta) \cdot E(X)\,] \leq e^{-E(X)\cdot\delta^2/2}$ for all $0 < \delta < 1$ (see e.g., [37]), we get that

$$\Pr[\, X \leq \ell/4\,] \leq \Pr[\, X \leq E(X)/2\,] \leq e^{-\frac{E(X)\cdot(1/2)^2}{2}} < e^{-\frac{\ell}{16}} = \frac{1}{e^{k\cdot\ln n}} = \frac{1}{n^k}.$$

By the union bound, since there are $\binom{n}{k} < n^k$ subsets of $[n]$ of size k, the probability that there exists a subset $A \subseteq [n]$ such that $|A| = k$ with at most $\ell/4$ functions h_i, for $i \in [\ell]$, such that $|h(A)| \neq k$, is less than 1. So, the probability that for every subset $A \subseteq [n]$ with $|A| = k$ there are more than $\ell/4$ hash functions h_i, for $i \in [\ell]$, such that $|h(A)| = k$ is greater than 0, and in particular a family H_{n,k,k^2} of $\ell = \Theta(k \cdot \log n)$ hash functions such as in the lemma exists. □

Using a family of perfect hash function H_{n,k,k^2} as in Lemma 4.5 and the scheme of Lemma 4.4 for every function in H_{n,k,k^2}, we get the a scheme in which the share size is $O(k^3 \cdot \log n)$ times the message size in the CDS protocol. Using Stinson's decomposition [41], we reduce the overhead.

Theorem 4.6. *Let Γ be a k-uniform access structure with n parties. Assume that for every k-input function $f : [N]^k \to \{0,1\}$ there is a k-party CDS protocol for f, for secrets of size t, in which the message size is $c(k, N, t)$. Then, for every $t' > \log n$, there is a secret-sharing scheme realizing Γ, for secrets of size $t = t' \cdot \Theta(k \cdot \log n)$, in which the share size of every party is $O(k^3 \cdot \log n \cdot c(k^2, n+1, t'))$.*

Proof. By Lemma 4.5, there exists a family of perfect hash functions $H_{n,k,k^2} = \{h_i : [n] \to [k^2] : i \in [\ell]\}$ with $\ell = \Theta(k \cdot \log n)$ functions, such that for every subset $A \subseteq [n]$ there is at least $\ell/4$ functions $h \in H_{n,k,k^2}$ for which $|h(A)| = k$. By Lemma 4.4, for every $i \in [\ell]$ there is a secret-sharing scheme Σ_{h_i} realizing the k-uniform access structure Γ_{h_i}, for secrets of size t, in which the share size of every party is $O(k^2 \cdot c(k^2, n+1, t))$. Also, by the definition of a family of perfect hash functions it holds that $\Gamma = \cup_{h \in H_{n,k,k^2}} \Gamma_h$.

To construct the desired secret-sharing scheme that realizes Γ, we use the Stinson's decomposition technique [41]. Let $\mathbb{F}$ be a finite field that contains at least $\max\{n, \ell\}$ elements. By an abuse of notation, we will assume that $\mathbb{F}$ is a prime field. Let $s = (s_1, \ldots, s_{\ell/4}) \in \mathbb{F}^{\ell/4}$ be the secret. We use a $(0, \ell/4)$-ramp secret-sharing scheme (that is, a scheme in which every set of size $\ell/4$ can reconstruct the secret, while there are no requirements on smaller sets) to generate shares $s_1, \ldots, s_\ell \in \mathbb{F}$ of s (that is, we choose a polynomial Q of degree $\ell/4 - 1$ such that $Q(i) = s_i$ for every $i \in [\ell/4]$ and define $s_i = Q(i)$ for every $i \in \{\ell/4 + 1, \ldots, \ell\}$). Then, for every $1 \leq i \leq \ell$, we independently generate shares of s_i using the scheme Σ_{h_i} that realizes the k-uniform access structure Γ_{h_i}, and give the shares to the parties in P. Since every set $A \subseteq P$ such that

$|A| = k$ satisfies $|h_i(A)| = k$ for at least $\ell/4$ values of $i \in [\ell]$, every authorized set $A \in \Gamma$ such that $|A| = k$ can reconstruct at least $\ell/4$ values from $s_1, \ldots, s_\ell$. Thus, by the property of the ramp scheme, the parties in A can reconstruct $s = (s_1, \ldots, s_{\ell/4})$.

Finally, let $t' = \log |\mathbb{F}|$. The combined scheme is a secret-sharing scheme that realizes the access structure Γ, in which the share size of every party is $\ell \cdot O(k^2 \cdot c(k^2, n+1, t')) = O(k^3 \cdot \log n \cdot c(k^2, n+1, t'))$. □

Remark 4.7. In the secret-sharing scheme of Theorem 4.6, if we start with a linear or multi-linear CDS protocol, then we result with a multi-linear secret-sharing scheme (i.e., a scheme in which the secret is a vector over a finite field $\mathbb{F}$, the random string is a vector over $\mathbb{F}$ chosen with uniform distribution, and each share is a vector over $\mathbb{F}$, where every element in the vector is a linear combination of the secret and the random elements).

Using the multi-linear CDS protocol of [2] for long secrets, in which the message size is $O(t)$, for secrets of size t (for big enough t), we get the following result.

Corollary 4.8. *Let Γ be a k-uniform access structure with n parties. Then, there is a multi-linear secret-sharing scheme realizing Γ, for secrets of size $t = \Omega(k \cdot \log n \cdot 2^{(n+1)^{k^2}})$, in which the share size of every party is $O(k^2 \cdot t)$.*

Remark 4.9. We can apply the transformation of Theorem 4.6 also to CDS protocols with short secrets. However, the best known k-party CDS protocol for such secrets of [36] (for k-input functions $f : [N]^k \to \{0,1\}$) have message size $2^{\tilde{O}(\sqrt{k \log N})}$, thus, using a k^2-party CDS would result in an inefficient secret-sharing scheme.

5 Optimal Linear Secret-Sharing Schemes Realizing k-Uniform Access Structures

In this section, we show how to construct a *linear* secret-sharing scheme realizing n-party k-uniform access structures in which the share size of every party is $O(n \cdot 2^{h(k/n)n/2})$. Using a result of [9], we prove a matching lower bound, which shows that our construction is optimal (up to a small polynomial factor).

We start by giving some high-level ideas of our linear secret-sharing scheme. We are inspired by the linear CDS protocols of [13], where for every Boolean n-input function they construct a linear CDS protocol with message size $O(2^{n/2})$ (a similar protocol with the same message size was independently constructed in [36]). By a transformation of [12], this implies that for every uniform access structure, there is a linear secret-sharing scheme with share size $O(n \cdot 2^{n/2})$. We want to optimize this construction for k-uniform access structures for $k < n/2$.

As a first step, we define balanced k-uniform access structures, where a k-uniform access structure is balanced if there exists a set of parties B of size $n/2$ such that every authorized set A of size k contains exactly $k/2$ parties in

B (that is, $|A \cap B| = k/2$ and $|A \setminus B| = k/2$). We construct an optimized secret-sharing scheme for balanced k-uniform access structures. We then show (using a probabilistic argument) that every k-uniform access structure Γ is a union of $O(k^{1/2} \cdot n)$ balanced k-uniform access structures. Thus, to realize Γ, we independently share the secret for each of the balanced access structures.

Definition 5.1 (The Access Structure Γ_B). *Let Γ be a k-uniform access structure with n parties for some even k and let $B \subseteq P$ be a subset of parties. The k-uniform access structure Γ_B is the access structure that contains all the subsets of parties of size greater than k and all authorized subsets from Γ of size k that contain exactly $k/2$ parties from the subset B. That is,*

$$\Gamma_B = \{A \subseteq P : |A| > k\} \cup \{A \subseteq P : A \in \Gamma, |A| = k, \text{ and } |A \cap B| = k/2\}\,.$$

Next, we show our basic linear scheme, which realizes the access structure Γ_B.

The secret: An element $s \in \mathbb{F}$, where $\mathbb{F}$ is a finite field.
The scheme: Assume without loss of generality that $B = \{P_1, \ldots, P_{n/2}\}$, and let $\mathcal{U} = \{U \subseteq B : |U| = k/2\}$ and $\mathcal{V} = \{V \subseteq \overline{B} : |V| = k/2\}$.

1. Share the secret s using a $(k+1)$-out-of-n secret-sharing scheme among the parties in P, that is, for every $j \in [n]$, give the jth share from this scheme to party P_j.
2. Choose two random elements $s_1, s_2 \in \mathbb{F}$, and let $s_3 = s_1 + s_2 + s$.
3. Share the element s_1 using a $k/2$-out-of-$n/2$ secret-sharing scheme among the parties in B, that is, for every $j \in [n/2]$, give the jth share from this scheme to party P_j.
4. Share the element s_2 using a $k/2$-out-of-$n/2$ secret-sharing scheme among the parties in $\overline{B}$, that is, for every $j \in [n/2]$, give the jth share from this scheme to party $P_{n/2+j}$.
5. For every $U \in \mathcal{U}$, choose a random element $r_U \in \mathbb{F}$.
6. For every $V = \left\{P_{j_1}, \ldots, P_{j_{k/2}}\right\} \in \mathcal{V}$, choose $k-1$ random elements $q_V^{j_1}, \ldots, q_V^{j_{k/2-1}} \in \mathbb{F}$.
7. Let $q_V = s_3 + \sum_{U \in \mathcal{U}: U \cup V \notin \Gamma} r_U$ and $q_V^{j_{k/2}} = q_V - (q_V^{j_1} + \cdots + q_V^{j_{k/2-1}})$.[a]
8. For every $j \in \{1, \ldots, n/2\}$, give the elements $(r_U)_{U \in \mathcal{U}: P_j \notin U}$ to P_j.
9. For every $j \in \{n/2+1, \ldots, n\}$, give the elements $(q_V^j)_{V \in \mathcal{V}: P_j \in V}$ to P_j.

[a] We assume that for every $V \in \mathcal{V}$ there exists a $U \in \mathcal{U}$ such that $U \cup V \notin \Gamma$; for example, this can be achieved by adding a "dummy" $U_0 \in \mathcal{U}$.

Fig. 3. A linear secret-sharing scheme Σ_B realizing the k-uniform access structure Γ_B.

Lemma 5.2. *Let $\mathbb{F}$ be a finite field and Γ be a k-uniform access structure with n parties for some even k and some even n, and let B be a subset of parties*

such that $|B| = n/2$*. Then, the scheme* Σ_B*, described in Fig. 3, is a linear secret-sharing scheme over* $\mathbb{F}$ *realizing* Γ_B *in which the share size of every party is* $O(k^{-1/2} \cdot 2^{h(k/n)n/2} \cdot \log|\mathbb{F}|)$.

Proof. Let $s \in \mathbb{F}$ be the secret. We prove that the secret-sharing scheme Σ_B is a linear scheme that realizes the k-uniform access structure Γ_B, with share size as in the lemma. First, we prove that in scheme Σ_B, every authorized set (i.e., a set in Γ_B) can reconstruct the elements s_1, s_2, and s_3, and, thus, it can reconstruct the secret s. Then, we prove that every unauthorized set (i.e., a set not in Γ_B) cannot learn any information about at least one of the field elements s_1, s_2, and s_3, and, thus, it cannot learn any information about the secret s.

Correctness. Every authorized set of size greater than k can reconstruct the secret using the $(k+1)$-out-of-n secret-sharing scheme.

For an authorized set $A \in \Gamma_B$ of size k, it holds that $|A \cap B| = k/2$. Thus, the parties in $A \cap B$ can reconstruct the element s_1 and the parties in $A \cap \overline{B}$ can reconstruct the element s_2 from the $k/2$-out-of-$n/2$ schemes. Additionally, the parties in $A \cap B$ hold the elements $(r_U)_{U \in \mathcal{U}}$ except for the element $r_{A \cap B}$ (since for every $U \in \mathcal{U}$ such that $U \neq A \cap B$ there exist at least one party from $A \cap B$ that is not in U) and the parties in $A \cap \overline{B}$ hold the elements $q^j_{A \cap \overline{B}}$ for every $P_j \in A \cap \overline{B}$, so they can reconstruct $q_{A \cap \overline{B}}$. Since $A = (A \cap B) \cup (A \cap \overline{B})$ is an authorized set then the element $r_{A \cap B}$ is not part of the sum computing $q_{A \cap \overline{B}}$. Thus, the parties in A can reconstruct the element s_3 and compute $s = s_1 + s_2 + s_3$.

Privacy. For an unauthorized set A of size less than k it holds that $|A \cap B| < k/2$ or $|A \cap \overline{B}| < k/2$, so the parties in A cannot learn any information about the secret s from the scheme Σ_B.

For an unauthorized set A of size k, if $|A \cap B| \neq k/2$, then $|A \cap B| < k/2$ or $|A \cap \overline{B}| < k/2$, so as above the parties in A cannot learn any information about the secret s from the scheme Σ_B.

Otherwise, if $|A \cap B| = k/2$, then $(A \cap B) \cup (A \cap \overline{B}) = A \notin \Gamma$, and the element $r_{A \cap B}$ is part of the sum computing $q_{A \cap \overline{B}}$, and the parties in A do not get the element $r_{A \cap B}$, so they cannot learn s_3 from $q_{A \cap \overline{B}}$. Similarly, for every $V = \{P_{j_1}, \ldots, P_{j_{k/2}}\} \in \mathcal{V}$ such that $V \neq A \cap \overline{B}$, there exists at least one party in $P_{j_i} \in V$ that is not in $A \cap \overline{B}$ (and, hence, not in A), so the parties in A do not get the element $q^{j_i}_V$. Thus, the parties in A cannot learn q_V, and, hence, they cannot learn s_3 from it.

Formally, a simulator Sim first applies the simulator of the $(k+1)$-out-of-n secret-sharing scheme for the set A and returns the output of this simulator. Then, Sim chooses random elements $s'_1, s'_2, s'_3 \in \mathbb{F}$, it shares s'_1 using a $k/2$-out-of-$n/2$ secret-sharing scheme, and for every $j \in [n/2]$ it returns the jth share from this scheme as part of the share of party P_j. Similarly, Sim shares s'_2 using a $k/2$-out-of-$n/2$ secret-sharing scheme, and for every $j \in [n/2]$ it returns the jth share from this scheme as part of the share of party $P_{n/2+j}$. Finally, for every $U \in \mathcal{U}$, the simulator Sim chooses random elements $r'_U \in \mathbb{F}$, and for every $V = \{P_{j_1}, \ldots, P_{j_{k/2}}\} \in \mathcal{V}$, it chooses $k-1$ random elements $q'^{j_1}_V, \ldots, q'^{j_{k/2-1}}_V \in \mathbb{F}$,

and lets

$$q'_V = s'_3 + \sum_{U\in\mathcal{U}:U\cup V\notin\Gamma} r'_U \qquad \text{and} \qquad q'^{j_{k/2}}_V = q'_V - (q'^{j_1}_V + \cdots + q'^{j_{k/2-1}}_V).$$

For every $j \in \{1,\ldots,n/2\}$ such that $P_j \in A$, Sim returns the elements $(r'_U)_{U\in\mathcal{U}:P_j\notin U}$ as part of the share of party P_j, and for every $j \in \{n/2+1,\ldots,n\}$ such that $P_j \in A$, it returns the elements $(q'^{j}_V)_{V\in\mathcal{V}:P_j\in V}$ as part of the share of party P_j.

Share Size. The share size of every party in the scheme Σ_B is

$$\begin{aligned} O((|\mathcal{U}|+|\mathcal{V}|)\cdot\log|\mathbb{F}| + \max\{\log|\mathbb{F}|,\log n\}) &= O\left(\binom{n/2}{k/2}\cdot\log|\mathbb{F}|\right) \\ &= O(k^{-1/2}\cdot 2^{h(k/n)n/2}\cdot\log|\mathbb{F}|), \end{aligned}$$

where the last equality holds by Fact 2.12. □

In the following we prove that a k-uniform access structure can be decomposed to $\ell = O(k^{1/2}\cdot n)$ balanced access structures.

Lemma 5.3. *Let P be a set of n parties for some even n, and let k be an even integer. Then, there are $\ell = \Theta(k^{1/2}\cdot n)$ subsets $B_1,\ldots,B_\ell \subseteq P$, each of them of size $n/2$, such that for every subset $A \subseteq P$ of size k it holds that $|A\cap B_i| = k/2$, for at least one $i \in [\ell]$.*

Proof. We choose at random a subset $B \subseteq P$ of size $n/2$, and for a given subset of parties $A \subseteq P$ of size k, we compute the probability that the size of $A\cap B$ is $k/2$. Note that this probability is the same as the probability that $|A\cap B| = k/2$ for a given subset of parties B of size $n/2$, and a random subset of parties A of size k, since this probability is the same for every subset $A \subseteq P$ of size k. The number of subsets A of size k is $\binom{n}{k}$. The number subsets A of size k such that $|A\cap B| = k/2$ is the number of options to choose $k/2$ parties from the $n/2$ parties from B times the number of options to choose $k/2$ parties from the $n/2$ parties from $\overline{B}$, which is $\binom{n/2}{k/2}^2$. Thus,

$$\begin{aligned} \Pr_{B\leftarrow\binom{P}{n/2}}[\,|A\cap B| = k/2\,] &= \Pr_{A\leftarrow\binom{P}{k}}[\,|A\cap B| = k/2\,] \\ &= \frac{\binom{n/2}{k/2}^2}{\binom{n}{k}} = \frac{\Theta\left(\left(\frac{1}{\sqrt{k}}\cdot 2^{h(k/n)n/2}\right)^2\right)}{\Theta\left(\frac{1}{\sqrt{k}}\cdot 2^{h(k/n)n}\right)} = \Theta\left(\frac{1}{k^{1/2}}\right). \end{aligned}$$

Hence, it holds that $\Pr[\,|A\cap B| \neq k/2\,] = 1 - \Theta\left(\frac{1}{k^{1/2}}\right)$.

We repeat the above process $\ell = \Theta(k^{1/2}\cdot n)$ times, and we get ℓ random subset of parties $B_1,\ldots,B_\ell$ of size $n/2$. By the union bound we get that

$$\begin{aligned} &\Pr[\,\exists A\subseteq[n], |A| = k : \forall B_i, i\in[\ell], |A\cap B_i| \neq k/2\,] \\ &\quad\le \binom{n}{k}\cdot(\Pr[\,|A\cap B| \neq k/2\,])^\ell \le 2^n\cdot\left(1-\Theta\left(\frac{1}{k^{1/2}}\right)\right)^\ell \le 2^n\cdot\frac{1}{e^n} < 1. \end{aligned}$$

Thus, the probability of choosing the desired subsets $B_1, \ldots, B_\ell$ is greater than 0, and in particular such subsets exist. □

Now, we are ready to present our final linear scheme, which realizes every k-uniform access structure.

Theorem 5.4. *Let Γ be a k-uniform access structure with n parties. Then, for every finite field $\mathbb{F}$, there is a linear secret-sharing scheme realizing Γ, for secrets from $\mathbb{F}$, in which the share size of every party is $O(n \cdot 2^{h(k/n)n/2} \cdot \log|\mathbb{F}|)$.*

Proof. Let $s \in \mathbb{F}$ be the secret. By adding dummy parties (which either belong to all authorized sets or belong to none of them), we can assume without loss of generality that k and n are even. By Lemma 5.3, there exist $\ell = \Theta(k^{1/2} \cdot n)$ subsets $B_1, \ldots, B_\ell \subseteq P$, where $|B_i| = n/2$ for every $i \in [\ell]$, such that for every subset $A \subseteq P$ of size k it holds that $|A \cap B_i| = k/2$ for at least one $i \in [\ell]$. Thus, we get that $\Gamma = \cup_{i=1}^{\ell} \Gamma_{B_i}$. By Lemma 5.2, for every $i \in [\ell]$ there is a linear secret-sharing scheme Σ_{B_i} realizing the k-uniform access structure Γ_{B_i}, in which the share size of every party is $O(k^{-1/2} \cdot 2^{h(k/n)n/2} \cdot \log|\mathbb{F}|)$. We independently realize every access structure Γ_{B_i} using the linear scheme Σ_{B_i} with secret s; the combined scheme is a linear secret-sharing scheme realizing the access structure Γ in which the share size of every party is

$$\ell \cdot O(k^{-1/2} \cdot 2^{h(k/n)n/2} \cdot \log|\mathbb{F}|) = O(n \cdot 2^{h(k/n)n/2} \cdot \log|\mathbb{F}|).$$

□

5.1 A Lower Bound for Linear Schemes Realizing k-Uniform Access Structures.

Using a result of [9] we prove a lower bound of $\tilde{O}(2^{h(k/n)n/2})$ on the share size of at least one party in every linear secret-sharing scheme that realizes k-uniform access structures, for one-bit secrets. As we have shown above for one-bit secrets (that is, $\mathbb{F} = \{0,1\}$), this bound is tight up to a $\mathrm{poly}(n)$ factor.

Theorem 5.5. *For most k-uniform access structures Γ with n parties, the share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2})$.*

Proof. If we share a one-bit secret using a linear secret-sharing scheme over $\mathbb{F}$ in which the largest share contains s field elements, then the size of the share of at least one party is $s \cdot \log|\mathbb{F}|$. For the share size of every party to be less than $k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}$, it must be that $|\mathbb{F}| \leq 2^{k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}}$ (otherwise, each share contains at least $k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}$ bits), and, obviously, $s \cdot \log|\mathbb{F}| \leq k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}$.

We say that the rank of an access structure Γ is r if the size of every minimal authorized set in Γ is at most r, so the rank of k-uniform access structures is $k+1$. By [9], for every finite field $\mathbb{F}$ and integers s, r, n such that $s > \log n$,

there are at most $2^{2rns^2 \cdot \log|\mathbb{F}|}$ access structures Γ with n parties and rank r such that there exists a linear secret-sharing scheme over $\mathbb{F}$ realizing Γ in which each share contains at most s field elements. Let $\theta = s \cdot \log|\mathbb{F}|$. Thus, there are at most $2^{2(k+1)n(\theta/\log|\mathbb{F}|)^2 \cdot \log|\mathbb{F}|} < 2^{2(k+1)n\theta^2}$ k-uniform access structures Γ with n parties such that there exists a linear secret-sharing scheme over $\mathbb{F}$ realizing Γ in which the share size of each party is at most θ.

Next, we count the number of linear schemes that realize k-uniform access structures in which the share size of each party is at most

$$\theta < k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}.$$

Since we are counting linear schemes, we need to sum the number of linear schemes that realizes k-uniform access structures for every possible finite field (there are at most $2^{k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}}$ such fields, because $|\mathbb{F}| \leq 2^{k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}}$). From all the above, the number of such linear schemes is at most

$$2^{k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2} + 2(k+1)n\theta^2}.$$

By Fact 2.12, the number of k-uniform access structures is $2^{\binom{n}{k}} = 2^{\Theta(k^{-1/2} \cdot 2^{h(k/n)n})}$. Thus, if half of the k-uniform access structures Γ with n parties have linear secret-sharing schemes in which the share size of every party is at most θ, then

$$2^{k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2} + 2(k+1)n\theta^2} \geq \frac{1}{2} \cdot 2^{\Theta(k^{-1/2} \cdot 2^{h(k/n)n})},$$

that is,

$$k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2} + 2(k+1)n\theta^2 \geq \Theta(k^{-1/2} \cdot 2^{h(k/n)n}),$$

and so

$$\theta = \Omega(k^{-3/4} \cdot n^{-1/2} \cdot 2^{h(k/n)n/2}).$$

□

6 Transformation from CDS to Secret-Sharing and Implications to Ad-hoc PSM

In this section, we describe a new transformation from a k-party CDS protocol to a secret-sharing scheme for k-uniform access structure. This construction improves the secret-sharing schemes for k-uniform access structures, for short secrets, compared to the scheme implied by the construction of [34]. We also show how to use the ideas of our transformation to construct a k-out-of-n ad-hoc PSM protocol from k-party PSM protocol.

6.1 The Transformation for Uniform Access Structures

We show how to realize any k-uniform access structure Γ with n parties using a k-party CDS protocol for the function g, defined in Definition 6.1.

Definition 6.1 (The Function g). *Let Γ be a k-uniform access structure with n parties. The k-input function $g : [n]^k \to \{0,1\}$ is the function that satisfies $g(x_1, \ldots, x_k) = 1$ if and only if $x_1 < \cdots < x_k$ and $A = \{P_{x_1}, \ldots, P_{x_k}\}$ is an authorized set, that is, $A \in \Gamma$.*

We say that a party P_{x_i} is the ith party in A if and only if there are $i-1$ parties before it and there are $k-i$ parties after it, when the indices of the parties are sorted. The idea of our scheme is that if party P_x is the ith party in a set A of size k, then its share will contain the message of the ith party in the CDS protocol for g (with the shared secret) when holding the input x. The problem with this idea is that the dealer does not know which set of parties will try to reconstruct the secret and it does not know if P_x is the ith party. If the dealer gives to two parties in the set the message of the ith party in the CDS protocol, then these parties get two different messages of the same party in the CDS protocol with different input, so we cannot ensure the privacy of the CDS protocol. Hence, some unauthorized sets may learn information about the secret. To solve this problem, the message of the ith party in the CDS protocol that party P_x gets will be masked by two random elements, such that only if P_x is the ith party in A, then the parties in A can learn this message. For this, the dealer shares one of the above mentioned random elements using a $(i-1)$-out-of-$(x-1)$ secret-sharing scheme and gives the shares to all parties before P_x, and shares the second random element using a $(k-i)$-out-of-$(n-x)$ secret-sharing scheme and gives the shares to all parties after P_x.

Theorem 6.2. *Let Γ be a k-uniform access structure with n parties, and assume that for every k-input function $f : [N]^k \to \{0,1\}$ there is a k-party CDS protocol for f for a one-bit secret, in which the message size is $c(k, N, 1)$. Then, the scheme Σ_g, described in Fig. 4, is a secret-sharing scheme realizing Γ, for a one-bit secret, in which the share size of every party is $O(k \cdot n \cdot c(k, n, 1))$.*

Proof. We prove that the secret-sharing scheme Σ_g is a scheme that realizes Γ with share size as in the theorem. Let $s \in \{0,1\}$ be the secret and $\mathcal{P}$ be a k-party CDS protocol for $g : [n]^k \to \{0,1\}$ (defined in Definition 6.1), for a one-bit secret, in which the message size is $c(k, n, 1)$. We prove that every subset of parties A of size k can learn only the messages corresponding to the parties in A of the CDS protocol for the function g (that is, party $P_j \in A$ can learn only the message $m_{i,j}$, where P_j is the ith party in A), so A can reconstruct the secret using these messages if and only if it is an authorized set. Additionally, we show that subsets of parties of size less than k cannot learn any messages of the CDS protocol for the function g, so such subsets cannot learn any information about the secret.

Correctness. An authorized set of size greater than k can reconstruct the secret using the $(k+1)$-out-of-n secret-sharing scheme.

The secret: An element $s \in \{0,1\}$.
The scheme: Let $g : [n]^k \to \{0,1\}$ be the k input function defined in Definition 6.1 and let $\mathcal{P}$ be a k-party CDS protocol for g.

1. Share the secret s using a $(k+1)$-out-of-n secret-sharing scheme among the parties in P, that is, for every $j \in [n]$, give the jth share from this scheme to party P_j.
2. Apply the k-party CDS protocol $\mathcal{P}$ to the k-input function g with the secret s and a common random string r that is chosen at random. For every $i \in [k]$, let $m_{i,x}$ be the message of the ith party in the CDS protocol $\mathcal{P}$ when holding the input $x \in [n]$.
3. For every $i \in \{2,\ldots,k\}$ and every $j \in \{i,\ldots,n-k+i\}$, choose a random string $q_{i,j}$ (of the same size as $m_{i,j}$), and for every $j \in \{1,\ldots,n-k+1\}$, let $q_{1,j} = \mathbf{0}$ (i.e., a string of zeroes).
4. For every $i \in \{1,\ldots,k-1\}$ and every $j \in \{i,\ldots,n-k+i\}$, choose a random string $r_{i,j}$ (of the same size as $m_{i,j}$), and for every $j \in \{k,\ldots,n\}$, let $r_{k,j} = \mathbf{0}$.
5. For every $i \in \{1,\ldots,k\}$ and every $j \in \{i,\ldots,n-k+i\}$, give the string $m_{i,j} \oplus r_{i,j} \oplus q_{i,j}$ to party P_j.
6. For every $i \in \{2,\ldots,k\}$ and every $j \in \{i,\ldots,n-k+i\}$, share the string $q_{i,j}$ using a $(i-1)$-out-of-$(j-1)$ secret-sharing scheme among the first $j-1$ parties (i.e., the parties $P_1,\ldots,P_{j-1}$), that is, for every $w \in [j-1]$, give the wth share from this scheme to party P_w.
7. For every $i \in \{1,\ldots,k-1\}$ and every $j \in \{i,\ldots,n-k+i\}$, share the string $r_{i,j}$ using a $(k-i)$-out-of-$(n-j)$ secret-sharing scheme among the last $n-j$ parties (i.e., the parties $P_{j+1},\ldots,P_n$), that is, for every $w \in [n-j]$, give the wth share from this scheme to party P_{j+w}.

Fig. 4. A secret-sharing scheme Σ_g realizing a k-uniform access structure Γ.

Let $A = \{P_{x_1},\ldots,P_{x_k}\}$ be an authorized set of size k such that $x_1 < \cdots < x_k$. For every $i \in [k]$, party P_{x_i} gets the string $m_{i,x_i} \oplus r_{i,x_i} \oplus q_{i,x_i}$. Additionally, the parties $P_{x_1},\ldots,P_{x_{i-1}}$ get $i-1$ shares from the $(i-1)$-out-of-(x_i-1) scheme for the string q_{i,x_i}, so they can reconstruct q_{i,x_i}, and the parties $P_{x_{i+1}},\ldots,P_{x_k}$ get $k-i$ shares from the $(k-i)$-out-of-$(n-x_i)$ scheme for the string r_{i,x_i}, so they can reconstruct r_{i,x_i}. Overall, for every $i \in [k]$, the parties $P_{x_1},\ldots,P_{x_k}$ learn the strings $m_{i,x_i} \oplus r_{i,x_i} \oplus q_{i,x_i}, r_{i,x_i}$, and q_{i,x_i}, so they can reconstruct the message m_{i,x_i} of the CDS protocol for g. Since $g(x_1,\ldots,x_k) = 1$, and the parties in A hold the messages $m_{1,x_1},\ldots,m_{k,x_k}$, they can reconstructs the secret s from those messages of the CDS protocol for g.

Privacy. Let $A = \{P_{x_1},\ldots,P_{x_k}\}$ be an unauthorized set of size k such that $x_1 < \cdots < x_k$. As claimed above, the parties in A can learn the messages $m_{1,x_1},\ldots,m_{k,x_k}$, but since $g(x_1,\ldots,x_k) = 0$, the parties in A cannot learn the secret from the CDS protocol for g (by the privacy of the CDS protocol).

We show that the parties in A cannot learn any other messages from the CDS protocol for g. For $x \in [n]$ such that $P_x \notin A$, the parties in A cannot learn $m_{i,x}$

for every $i \in [k]$, since they do not get this message (even masked by random strings). Consider an $x \in [n]$ such that $P_x \in A$ and $x \neq x_i$ for some $i \in [k]$. If $x < x_i$ (that is, P_x is smaller than the ith party in A) then the parties in A cannot learn the string $q_{i,x}$, since they hold less than $i-1$ shares from the $(i-1)$-out-of-$(x-1)$ for the string $q_{i,x}$, so the parties in A cannot learn the message $m_{i,x}$. Otherwise, if $x > x_i$ (that is, P_x is bigger than the ith party in A) then the parties in A cannot learn the string $r_{i,x}$, since they hold less than $k-i$ shares from the $(k-i)$-out-of-$(n-x)$ for the string $r_{i,x}$, so the parties in A cannot learn the message $m_{i,x}$. Thus, the parties in A cannot learn any information about the secret s.

The last argument holds for unauthorized sets A of size less than k, so such sets A cannot learn any messages from the CDS protocol for g, and, thus, cannot learn any information about the secret s.

Formally, a simulator SIM first applies the simulator of the $(k+1)$-out-of-n secret-sharing scheme for a one-bite secret with the set A and returns the output of this simulator. Then, SIM applies the simulator of the CDS protocol $\mathcal{P}$ with inputs $x_1, \ldots, x_k$; let m'_i be the output of the simulator of $\mathcal{P}$ for the message of the ith party in $\mathcal{P}$ (when holding the input x_i). Then, for every $i \in \{2, \ldots, k\}$ and every $j \in \{i, \ldots, n-k+i\}$, the simulator SIM chooses a random string $q'_{i,j}$, and for every $i \in \{1, \ldots, k-1\}$ and every $j \in \{i, \ldots, n-k+i\}$, the simulator SIM chooses a random string $r'_{i,j}$. For every $j \in \{1, \ldots, n-k+1\}$, let $q'_{1,j} = \mathbf{0}$, and for every $j \in \{k, \ldots, n\}$, let $r'_{k,j} = \mathbf{0}$. For every $i \in \{1, \ldots, k\}$ the simulator SIM returns $m'_i \oplus r'_{i,x_i} \oplus q'_{i,x_i}$ and $(r'_{i,j} \oplus q'_{i,j})_{j \neq x_i}$ as part of the share of party P_{x_i}. For every $i \in \{2, \ldots, k\}$ and every $j \in \{i, \ldots, n-k+i\}$, the simulator SIM shares $q'_{i,j}$ using an $(i-1)$-out-of-$(j-1)$ secret-sharing scheme, and for every $w \in [j-1]$ such that $P_w \in A$, it returns the wth share from this scheme as part of the share of party P_w. For every $i \in \{1, \ldots, k-1\}$ and every $j \in \{i, \ldots, n-k+i\}$, the simulator SIM shares $r'_{i,j}$ using a $(k-i)$-out-of-$(n-j)$ secret-sharing scheme, and for every $w \in [n-j]$ such that $P_{j+w} \in A$, it returns the wth share from this scheme as part of the share of party P_{j+w}.

SHARE SIZE. The share size of every party in the scheme Σ_h is

$$O(k \cdot n \cdot c(k, n, 1) + \log n) = O(k \cdot n \cdot c(k, n, 1)).$$

□

Using the CDS protocol of [36], in which the message size is $2^{O(\sqrt{k \log n} \log(k \log n))}$, for a one-bit secret, we get the following result.

Corollary 6.3. *Let Γ be a k-uniform access structure with n parties. Then, there is a secret-sharing scheme realizing Γ, for a one-bit secret, in which the share size of every party is $k \cdot n \cdot 2^{O(\sqrt{k \log n} \log(k \log n))}$.*

6.2 The Transformation for Ad-hoc PSM Protocols

We use the same ideas as in the above transformation to construct a k-out-of-n ad-hoc PSM protocol for a function $f : [N]^k \to Y$ using a k-party PSM protocol

for f. Recall that some k parties $P_{i_1}, \ldots, P_{i_k}$, holding inputs $x_{i_1}, \ldots, x_{i_k} \in [N]$ respectively, participate in the protocol, and they want to compute $f(x_{i_1}, \ldots, x_{i_k})$. However, the participating parties do not know which k parties among the n parties participate in the protocol. In Fig. 5, we describe our ad-hoc PSM protocol; in the protocol there is an offline stage, which contains computation that only depends on the common string, and an online stage in which each participating party sends its message.

Offline stage of the protocol: Let $\mathcal{P}$ be a k-party PSM protocol for f.

1. Apply the k-party PSM protocol $\mathcal{P}$ for the k-input function f with a common random string r that is chosen at random. For every $i \in [k]$ let $m_{i,x}$ be the message of the ith party in the PSM protocol $\mathcal{P}$ when holding the input $x \in [N]$.
2. For every $i \in \{2, \ldots, k\}$ and every $j \in \{i, \ldots, n-k+i\}$, choose a random string $q_{i,j}$ (of the same size as $m_{i,j}$), and for every $j \in \{1, \ldots, n-k+1\}$, let $q_{1,j} = \mathbf{0}$ (i.e., a string of zeroes).
3. For every $i \in \{1, \ldots, k-1\}$ and every $j \in \{i, \ldots, n-k+i\}$, choose a random string $r_{i,j}$ (of the same size as $m_{i,j}$), and for every $j \in \{k, \ldots, n\}$, let $r_{k,j} = \mathbf{0}$ (i.e., a string of zeroes).
4. For every $i \in \{2, \ldots, k\}$ and every $j \in \{i, \ldots, n-k+i\}$, share the string $q_{i,j}$ using a $(i-1)$-out-of-$(j-1)$ secret-sharing scheme among the first $j-1$ parties (i.e., the parties $P_1, \ldots, P_{j-1}$). For every $w \in \{1, \ldots, j-1\}$, let $q^w_{i,j}$ be the share of party P_w.
5. For every $i \in \{1, \ldots, k\}$ and every $j \in \{i, \ldots, n-k+i\}$, share the string $r_{i,j}$ using a $(k-i)$-out-of-$(n-j)$ secret-sharing scheme among the last $n-j$ parties (i.e., the parties $P_{j+1}, \ldots, P_n$). For every $w \in \{j+1, \ldots, n\}$, let $r^w_{i,j}$ be the share of party P_w.

Online stage of the protocol for a set A of k parties: Each party $P_j \in A$ holds an input $x_j \in [N]$.

1. Every party $P_j \in A$ sends to the referee the string $m_{i,x_j} \oplus r_{i,j} \oplus q_{i,j}$ for every $i \in \{1, \ldots, k\}$.
2. Every party $P_w \in A$ sends to the referee the string $q^w_{i,j}$ for every $i \in \{1, \ldots, k\}$ and every $j > w$.
3. Every party $P_w \in A$ sends to the referee the string $r^w_{i,j}$ for every $i \in \{1, \ldots, k\}$ and every $j < w$.

Fig. 5. A k-out-of-n ad-hoc PSM protocol $\mathcal{P}_f$ for a k-input function $f : [N]^k \to Y$.

Theorem 6.4. *Let $f : [N]^k \to Y$ be a k-input function, for some integer k, and assume that there is a k-party PSM protocol for f with message size $c_f(k, N)$. Then, the protocol $\mathcal{P}_f$, described in Fig. 5, is a k-out-of-n ad-hoc PSM protocol for f with message size $O(k \cdot n \cdot c_f(k, N))$.*

Proof. The correctness of the protocol follows from the fact that given k parties $P_{i_1}, \ldots, P_{i_k}$, the referee learns the messages $m_{1,x_{i_1}}, \ldots, m_{k,x_{i_k}}$, as explained in the proof of Theorem 6.2, and thus, by the correctness of the PSM protocol for f, the referee can learn $f(x_{i_1}, \ldots, x_{i_k})$. The privacy of the protocol follows from the privacy of the PSM protocol and the fact that the referee learns only the messages $m_{1,x_{i_1}}, \ldots, m_{k,x_{i_k}}$, as proved in Theorem 6.2. Note that for less than k parties, the referee cannot learn any message of the PSM protocol, again like in Theorem 6.2. □

By the PSM protocol of [12], in which the message size is $O(k^3 \cdot N^{k/2})$, we get the following result.

Corollary 6.5. *Let $f : [N]^k \to Y$ be a k-input function, for some integer k. Then, there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(k^4 \cdot n \cdot N^{k/2})$.*

6.3 Improving the Ad-hoc PSM Protocol for Symmetric Functions

We combine the protocol of Section 6.2 with the ideas of Section 4, and construct a better k-out-of-n ad-hoc PSM protocol for symmetric functions $f : [N]^k \to Y$, where a function f is symmetric if for a given input $x = (x_1, \ldots, x_k)$, the output of f on the input x is the same as the output of f on any permutation on the order of the x_i's, that is, for every $x = (x_1, \ldots, x_k)$ and every permutation $\pi : [k] \to [k]$, it holds that $f(x_1, \ldots, x_k) = f(x_{\pi(1)}, \ldots, x_{\pi(k)})$.

Our construction consists of two steps. First, we show that we can construct a k-out-of-n ad-hoc PSM protocol for f using $\Theta(k \cdot \log n)$ invocations of a k-out-of-k^2 ad-hoc PSM protocol. Then, we use the protocol of Theorem 6.4 with k^2 parties, and get a k-out-of-n ad-hoc PSM protocol for f with message size $O(k^4 \cdot \log n \cdot c_f(k, N))$, where $c_f(k, N)$ is the message size of a k-party PSM protocol for f.

For the first step, we show a general transformation from k-out-of-t ad-hoc PSM protocols to k-out-of-n ad-hoc PSM protocols, for every $k \leq t \leq n$. This transformation generalizes and improves the construction of [11], which only works when $t = k$. As mentioned above, we use this transformation for $t = k^2$. For our transformation, we take a family of perfect hash functions $H_{n,k,t}$, and construct a k-out-of-n ad-hoc PSM protocol for f using independent copies of a k-out-of-t ad-hoc PSM protocol for f, one copy for each hash function $h \in H_{n,k,t}$.

In the k-out-of-t ad-hoc PSM protocol for h, denoted by $\mathcal{P}_h$, party P_j simulates the $h(j)$th party in $\mathcal{P}_h$. If h is one-to-one on a set of k parties, that is, the set does not collide on h, then the referee gets k messages of k different parties of the protocol $\mathcal{P}_h$, so it can compute the output of f on the inputs of the parties.

If a set collides on h, then the referee gets at least two messages of the same party of the protocol $\mathcal{P}_h$, so the privacy is not guaranteed. To solve this problem, every party encrypts its message of the protocol $\mathcal{P}_h$ using an information-theoretic encryption system that is secure as long as the adversary sees at most k encryptions. We also share the encryption key using a k-out-of-t secret-sharing

scheme, and party P_j sends to the referee the $h(j)$th share from this scheme. For sets of size less than k and sets of size k that collide on h, the referee cannot reconstruct the key and sees at most k encrypted messages, thus cannot learn any information on the messages of the protocol $\mathcal{P}_h$. For the encryption system, we use a polynomial of degree k as the encryption key; to encrypt a message each party masks it by a unique point of the polynomial.

Observe that the referee might learn the output of f from more than one protocol, for several functions from $H_{n,k,t}$, so the requirement for symmetric functions is necessary, since the order of the parties in a set of size k can change according to the different hash functions.

Offline stage of the protocol: Let $\mathcal{P}$ be a k-out-of-t ad-hoc PSM protocol for $f : [N]^k \to Y$ and let $h : [n] \to [t]$ be a hash function.

1. Apply the k-out-of-t ad-hoc PSM protocol $\mathcal{P}$ for the k-input function f with a common random string r chosen at random with uniform distribution. For every $i \in [t]$ let $m_{i,x}$ be the message of the ith party in the ad-hoc PSM protocol $\mathcal{P}$ when holding the input $x \in [N]$.
2. Choose a random polynomial Q of degree k over a finite field $\mathbb{F}$ such that $\log |\mathbb{F}| > \max\{\log n, c_f(k,t,N)\}$.
3. Share the polynomial Q (i.e., its coefficients) using a k-out-of-t secret-sharing scheme. For every $i \in \{1,\ldots,t\}$, let q^i be the ith share from this scheme.

Online stage of the protocol for a set A of k parties: Each party $P_j \in A$, who holds an input $x_j \in [N]$, sends $m_{h(j),x_j} \oplus Q(j)$ and $q^{h(j)}$ to the referee.

Fig. 6. A k-out-of-n ad-hoc PSM protocol $\mathcal{P}_h$ for a symmetric k-input function $f : [N]^k \to Y$.

Lemma 6.6. *Let $f : [N]^k \to Y$ be a k-input symmetric function, for some integer k, and assume that there is a k-out-of-t ad-hoc PSM protocol $\mathcal{P}$ for f with message size $c_f(k,t,N)$, and that there is a family of perfect hash function $H_{n,k,t} = \{h_i : [n] \to [t] : i \in [\ell]\}$. Then, there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(\ell \cdot k \cdot \max\{c_f(k,t,N), \log n\})$.*

Proof. We show that the protocol that consist of the ℓ independent k-out-of-t ad-hoc PSM protocols $\mathcal{P}_h$, described in Fig. 6, for every $h \in H_{n,k,t}$, is a k-out-of-n ad-hoc PSM protocol. We prove that for every subset of k parties that holds the inputs $x_{i_1},\ldots,x_{i_k} \in [N]$, the referee can compute $f(x_{i_1},\ldots,x_{i_k})$. Next, we show that for every subset of k parties or less, the referee cannot learn any additional information (except for the output of f for subsets of k parties) about the inputs of the parties.

Correctness. Let $A = \{P_{i_1},\ldots,P_{i_k}\} \subseteq P$ be a subset of k parties. Thus, there exist a function $h \in H_{n,k,t}$ such that $|\{h(i_1),\ldots,h(i_k)\}| = k$. In the protocol

$\mathcal{P}_h$ the referee can reconstruct the polynomial Q, since it gets k distinct shares of the k-out-of-t secret-sharing scheme for Q. Thus, the referee can compute $Q(i_j)$ for every $j \in [k]$. Additionally, for every $j \in [k]$, party $P_{i_j} \in A$ sends to the referee the message $m_{h(i_j),x_{i_j}} \oplus Q(i_j)$. Overall, for every $j \in [k]$, the referee has the strings $m_{h(i_j),x_{i_j}} \oplus Q(i_j)$ and $Q(i_j)$, so it can reconstruct the message $m_{h(i_j),x_{i_j}}$ of the k-out-of-t ad-hoc PSM protocol $\mathcal{P}$ for f. By the correctness of the protocol $\mathcal{P}$, the referee can compute $f(x_{i_1}, \ldots, x_{i_k})$.

Privacy. Consider a subset $A = \{P_{i_1}, \ldots, P_{i_{k'}}\} \subseteq P$ of $k' \leq k$ parties and a function $h \in H_{n,k,t}$. If $|\{h(i_1), \ldots, h(i_{k'})\}| < k$ then the referee cannot learn the polynomial Q, since it gets less than k distinct shares of the k-out-of-t secret-sharing scheme for Q. Furthermore, since the degree of Q is k, the $k' \leq k$ points $Q(i_1), \ldots, Q(i_{k'})$ are uniformly distributed, i.e., the encrypted messages $m_{h(i_1),x_{i_1}} \oplus Q(i_1), \ldots, m_{h(i_{k'}),x_{i_{k'}}} \oplus Q(i_{k'})$ are uniformly distributed. So the referee cannot learn any information on the message of the ad-hoc PSM protocol $\mathcal{P}$ from the messages of $\mathcal{P}_h$. If $|\{h(i_1), \ldots, h(i_{k'})\}| = k$ (in particular $k' = k$) then the referee learns exactly the above mentioned messages $m_{h(i_j),x_{i_j}}$ for every $j \in [k]$ (i.e., k messages of distinct k parties in the protocol $\mathcal{P}$), and by the privacy of the ad-hoc PSM protocol $\mathcal{P}$ that uses h the referee cannot learn any additional information (not implied by $f(x_{i_1}, \ldots, x_{i_1})$) about the inputs of the parties from the messages it gets in this protocol. Thus, since for every $h \in H_{n,k,k^2}$ the k-out-of-t ad-hoc PSM protocol $\mathcal{P}$ that uses h is private, and since those protocols are independent, we get that the resulting k-out-of-n ad-hoc PSM protocol is also private.

Message Size. The message of each party in the protocol $\mathcal{P}_h$ (for some $h \in H_{n,k,t}$) contains one message of $\mathcal{P}$ and one share Q, a polynomial of degree k over a field $\mathbb{F}$ such that $\log |\mathbb{F}| \approx \max\{\log n, c_f(k,t,N)\}$, thus the message size is $O(k \max\{\log n, c_f(k,t,N)\})$. The message size of the resulting k-out-of-n ad-hoc PSM protocol for f is $O(\ell \cdot k \cdot \max\{\log n, c_f(k,t,N)\})$. □

By taking $t = k^2$ and using our ad-hoc PSM protocol from Theorem 6.4 and the family of perfect hash functions from Lemma 4.5, we get the following result.

Theorem 6.7. *Let $f : [N]^k \to Y$ be a k-input symmetric function, for some integer k, and assume that there is a k-party PSM protocol for f with message size $c_f(k,N)$. Then, there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(k^5 \cdot \log n \cdot c_f(k,N))$.*

Proof. By Theorem 6.4, there is a k-out-of-k^2 ad-hoc PSM protocol for f with message size $O(k \cdot k^2 \cdot c_f(k,N)) = O(k^3 \cdot c_f(k,N))$, and by Lemma 4.5, there is a family of perfect hash functions H_{n,k,k^2} with $\ell = \Theta(k \cdot \log n)$ functions.

Thus, by Lemma 6.6, there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(\ell \cdot k \cdot k^3 \cdot c_f(k,N)) = O(k^5 \cdot \log n \cdot c_f(k,N))$. □

Finally, again by the PSM protocol of [12], we obtain the next result.

Corollary 6.8. *Let $f : [N]^k \to Y$ be a k-input symmetric function, for some integer k. Then, there is a k-out-of-n ad-hoc PSM protocol for f with message size $O(k^8 \cdot \log n \cdot N^{k/2})$.*

Acknowledgement. The first and fourth authors are supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, and the Check Point Institute for Information Security. Part of this work was done while the second author was visiting Georgetown university, supported by NSF grant no. 1565387, TWC: Large: Collaborative: Computing Over Distributed Sensitive Data. The second author is also supported by ISF grant 152/17 and by a grant from the Cyber Security Research Center at Ben-Gurion University of the Negev. The third author is supported by the Spanish Government through TIN2014-57364-C2-1-R and by the Government of Catalonia through Grant 2017 SGR 705. The fifth author is supported by ISF grant 152/17, by a grant from the Cyber Security Research Center at Ben-Gurion University of the Negev, and by the Frankel center for computer science.

References

1. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: EUROCRYPT 2001. LNCS, vol. 2045, pp. 118–134 (2001)
2. Applebaum, B., Arkis, B.: On the power of amortization in secret sharing: d-uniform secret sharing and CDS with constant information rate. In: TCC 2018. LNCS, vol. 11239, pp. 317–344. Springer-Verlag (2018)
3. Applebaum, B., Arkis, B., Raykov, P., Vasudevan, P.N.: Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. In: CRYPTO 2017. LNCS, vol. 10401, pp. 727–757 (2017)
4. Applebaum, B., Holenstein, T., Mishra, M., Shayevitz, O.: The communication complexity of private simultaneous messages, revisited. In: EUROCRYPT 2018. pp. 261–286. LNCS (2018)
5. Applebaum, B., Vasudevan, P.: Placing conditional disclosure of secrets in the communication complexity universe. In: 10th Innovations in Theoretical Computer Science Conference, ITCS. LIPIcs, vol. 124, pp. 4:1–4:14 (2019)
6. Applebaum, B., Beimel, A., Farràs, O., Nir, O., Peter, N.: Secret-sharing schemes for general and uniform access structures. LNCS, vol. 11478, pp. 441–471 (2019)
7. Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577 (2014)
8. Beimel, A., Chor, B.: Universally ideal secret-sharing schemes. IEEE Trans. on Information Theory 40(3), 786–794 (1994)
9. Beimel, A., Farràs, O., Mintz, Y., Peter, N.: Linear secret-sharing schemes for forbidden graph access structures. In: TCC 2017. LNCS, vol. 10678, pp. 394–423 (2017)
10. Beimel, A., Farràs, O., Peter, N.: Secret sharing schemes for dense forbidden graphs. In: SCN 2016. pp. 509–528 (2016)
11. Beimel, A., Ishai, Y., Kumaresan, R., Kushilevitz, E.: On the cryptographic complexity of the worst functions. In: TCC 2014. LNCS, vol. 8349, pp. 317–342 (2014)
12. Beimel, A., Kushilevitz, E., Nissim, P.: The complexity of multiparty PSM protocols and related models. In: EUROCRYPT 2018. pp. 287–318. LNCS (2018)
13. Beimel, A., Peter, N.: Optimal linear multiparty conditional disclosure of secrets protocols. In: ASIACRYPT 2018. LNCS, vol. 11274, pp. 332–362 (2018)

14. Beimel, A., Ishai, Y., Kushilevitz, E.: Ad hoc PSM protocols: Secure computation without coordination. In: EUROCRYPT 2017. LNCS, vol. 10212, pp. 580–608 (2017)
15. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for noncryptographic fault-tolerant distributed computations. In: Proc. of the 20th ACM Symp. on the Theory of Computing. pp. 1–10 (1988)
16. Benaloh, J., Leichter, J.: Generalized secret sharing and monotone functions. In: CRYPTO '88. LNCS, vol. 403, pp. 27–35 (1990)
17. Bertilsson, M., Ingemarsson, I.: A construction of practical secret sharing schemes using linear block codes. In: AUSCRYPT '92. LNCS, vol. 718, pp. 67–79 (1993)
18. Blakley, G.R.: Safeguarding cryptographic keys. In: Proc. of the 1979 AFIPS National Computer Conference. vol. 48, pp. 313–317 (1979)
19. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols. In: Proc. of the 20th ACM Symp. on the Theory of Computing. pp. 11–19 (1988)
20. Chor, B., Kushilevitz, E.: Secret sharing over infinite domains. J. of Cryptology 6(2), 87–96 (1993)
21. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334 (2000)
22. Csirmaz, L.: The size of a share must be large. In: EUROCRYPT '94. LNCS, vol. 950, pp. 13–22 (1995), journal version in: *J. of Cryptology*, 10(4):223–231, 1997.
23. Csirmaz, L.: The dealer's random bits in perfect secret sharing schemes. Studia Sci. Math. Hungar. 32(3–4), 429–437 (1996)
24. Desmedt, Y., Frankel, Y.: Shared generation of authenticators and signatures. In: CRYPTO '91. LNCS, vol. 576, pp. 457–469 (1992)
25. Erdös, P., Spencer, J.: Probabilistic Methods in Combinatorics. Academic Press (1974)
26. Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation. In: 26th STOC 1994. pp. 554–563 (1994)
27. Gay, R., Kerenidis, I., Wee, H.: Communication complexity of conditional disclosure of secrets and attribute-based encryption. In: CRYPTO 2015. LNCS, vol. 9216, pp. 485–502 (2015)
28. Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. J. of Computer and System Sciences 60(3), 592–629 (2000)
29. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proc. of the 13th ACM conference on Computer and communications security. pp. 89–98 (2006)
30. Ishai, Y., Kushilevitz, E.: Private simultaneous messages protocols with applications. In: 5th Israel Symp. on Theory of Computing and Systems. pp. 174–183 (1997)
31. Ishai, Y., Wee, H.: Partial garbling schemes and their applications. In: 41st ICALP. vol. 8572, pp. 650–662 (2014)
32. Ito, M., Saito, A., Nishizeki, T.: Secret sharing schemes realizing general access structure. In: Globecom 87. pp. 99–102 (1987), Journal version: Multiple assignment scheme for sharing secret. J. of Cryptology 6(1), 15-20, (1993).
33. Karchmer, M., Wigderson, A.: On span programs. In: 8th Structure in Complexity Theory. pp. 102–111 (1993)
34. Liu, T., Vaikuntanathan, V.: Breaking the circuit-size barrier in secret sharing. In: 50th STOC 2018. pp. 699–708 (2018)

35. Liu, T., Vaikuntanathan, V., Wee, H.: Conditional disclosure of secrets via non-linear reconstruction. In: CRYPTO 2017. LNCS, vol. 10401, pp. 758–790 (2017)
36. Liu, T., Vaikuntanathan, V., Wee, H.: Towards breaking the exponential barrier for general secret sharing. In: EUROCRYPT 2018. pp. 567–596. LNCS (2018)
37. Mitzenmacher, M., Upfal, E.: Probability and Computing. Cambridge University Press (2005)
38. Naor, M., Wool, A.: Access control and signatures via quorum secret sharing. In: 3rd ACM Conf. on Computer and Communications Security. pp. 157–167 (1996)
39. Shamir, A.: How to share a secret. Communications of the ACM 22, 612–613 (1979)
40. Shankar, B., Srinathan, K., Rangan, C.P.: Alternative protocols for generalized oblivious transfer. In: Proc. of ICDCN 2008. LNCS, vol. 4904, pp. 304–309 (2008)
41. Stinson, D.R.: Decomposition construction for secret sharing schemes. IEEE Trans. on Information Theory 40(1), 118–125 (1994)
42. Sun, H., Shieh, S.: Secret sharing in graph-based prohibited structures. In: INFOCOM '97. pp. 718–724 (1997)
43. Tassa, T.: Generalized oblivious transfer by secret sharing. Designs, Codes and Cryptography 58(1), 11–21 (2011)
44. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: PKC 2011. LNCS, vol. 6571, pp. 53–70 (2011)
45. Wee, H.: Dual system encryption via predicate encodings. In: TCC 2014. LNCS, vol. 8349, pp. 616–637 (2014)