Optimal Linear Multiparty Conditional Disclosure of Secrets Protocols*

Amos Beimel and Naty Peter

Ben-Gurion University of the Negev, Be'er-Sheva, Israel
amos.beimel@gmail.com, naty@post.bgu.ac.il

Abstract. In a k-party CDS protocol, each party sends one message to a referee (without seeing the other messages) such that the referee will learn a secret held by the parties if and only if the inputs of the parties satisfy some condition (e.g., if the inputs are all equal). This simple primitive is used to construct attribute based encryption, symmetrically-private information retrieval, priced oblivious transfer, and secret-sharing schemes for any access structure. Motivated by these applications, CDS protocols have been recently studied in many papers.

In this work, we study linear CDS protocols, where each of the messages of the parties is a linear function of the secret and random elements taken from some finite field. Linearity is an important property of CDS protocols as many applications of CDS protocols required it.

Our main result is a construction of linear k-party CDS protocols for an arbitrary function $f : [N]^k \to \{0,1\}$ with messages of size $O(N^{(k-1)/2})$.[1] By a lower bound of Beimel et al. [TCC 2017], this message size is optimal. We also consider functions with few inputs that return 1, and design more efficient CDS protocols for them.

CDS protocols can be used to construct secret-sharing schemes for uniform access structures, where for some k all sets of size less than k are unauthorized, all sets of size greater than k are authorized, and each set of size k can be either authorized or unauthorized. We show that our results imply that every k-uniform access structure with n parties can be realized by a linear secret-sharing scheme with share size $\min\left\{(O(n/k))^{(k-1)/2}, O(n \cdot 2^{n/2})\right\}$. Furthermore, the linear k-party CDS protocol with messages of size $O(N^{(k-1)/2})$ was recently used by Liu and Vaikuntanathan [STOC 2018] to construct a linear secret-sharing scheme with share size $O(2^{0.999n})$ for any n-party access structure.

Key words. Secret-sharing schemes, conditional disclosure of secrets protocols.

1 Introduction

Conditional disclosure of secrets (CDS) protocols, introduced by Gertner, Ishai, Kushilevitz, and Malkin [20], is a cryptographic primitive related to secret-

* The authors are supported by ISF grant 152/17 and by the Frankel center for computer science.

[1] A similar result was independently and in parallel proven by Liu et al. [27].

sharing that has many applications. In a CDS protocol, there are k parties, each one holds a private input x_i and the same secret s, and a referee that holds $x_1, \ldots, x_k$ but does know s. The goal is that the referee will learn s if and only if the inputs $x_1, \ldots, x_k$ satisfy some condition specified by a function f, i.e., $f(x_1, \ldots, x_k) = 1$. The challenge is that each party sends only one message to the referee (without seeing the other messages). This simple primitive is used to construct attribute based encryption [6, 29], symmetrically-private information retrieval [20], priced oblivious transfer [1], secret-sharing for uniform access structures [13, 3, 14], and secret-sharing for general access structures [25]. Motivated by these applications, CDS protocols have been recently studied in many papers [22, 19, 12, 4, 10, 26, 3, 27, 14].

In this work, we study linear CDS protocols, where the messages of the parties are a linear function of the secret and random elements taken from some finite field. Equivalently, a CDS protocol is linear if the reconstruction of the secret by the referee from the messages is a linear mapping.[2] In many applications of CDS protocols, it is required that the protocol will be linear. For example, it was shown by Attrapadung [6] and Wee [29] that linear 2-party CDS protocols can be used to construct public-key (multi-user) attribute-based encryption. Furthermore, using a construction of Cramer et al. [16] and the construction of secret-sharing schemes of [25], linear k-party CDS protocols imply secure multi-party computation (MPC) protocols secure against Q2 adversarial structures.[3] The construction of Cramer et al. [16] requires a linear secret-sharing scheme, i.e., they must use a linear k-party CDS.

Linear CDS protocols can be used to construct linear secret-sharing schemes for uniform access structures, that is, access structures in which for some k all sets of size less than k are unauthorized, all sets of size greater than k are authorized, and each set of size k can be either authorized or unauthorized [13, 12, 3, 14]. Very recently, Liu et al. [25] used the optimal linear k-party CDS protocols (constructed in our paper and in [27]) to construct linear secret-sharing schemes with share size $O(2^{0.999n})$ for any n-party access structure. They also used non-linear k-party CDS protocols to construct a non-linear secret-sharing scheme with share size $O(2^{0.994n})$ for any n-party access structure. These are the first major improvements in the share size of secret-sharing schemes for arbitrary access structures since the first constructions of [23], whose share size is 2^n.

CDS protocols share similarities with private simultaneous messages (PSM) protocols, a primitive introduced by Feige, Kilian, and Naor [18] for two-input functions, and generalized to k-input functions in [18, 21]. In a PSM protocol, there are k parties, each one holds a private input x_i; here the referee does not hold $x_1, \ldots, x_k$. The goal is that the referee will learn $f(x_1, \ldots, x_k)$, without learning any additional information on $x_1, \ldots, x_k$. As in CDS protocols, the challenge is that each party sends only one message to the referee (without seeing

[2] This equivalence is a special case of the equivalence for secret-sharing schemes. See [7] for discussion on equivalent definitions of linear secret-sharing schemes.

[3] An adversarial structure is Q2 if the union of any two sets that the adversary can control is not the entire set of parties.

the other messages). Intuitively, compared to CDS, PSM is a stronger model, since in CDS the inputs are known to the referee and in PSM the referee should not learn any information about the inputs. A PSM protocol for a function f implies a CDS protocol for the function f [20]. PSM protocols for specific functions are used in the construction of CDS protocols in our work and in [27].

1.1 Our Results

Our first result is a construction of linear k-party CDS protocols for an arbitrary k-input function $f : [M] \times [N]^{k-1} \to \{0,1\}$ with total message size $O(N^{(k-1)/2})$ for every $k > 2$ and integers M, N. Notice that the message size is independent of M, that is, the domain of inputs of one party can be very large without affecting the message size. For example, this property is useful for the index function where the size of the domain of the first party is $2^{N^{k-1}}$ and the size of the domains of the other parties in N. By [10], the size of the messages in linear CDS protocols for most k-input functions $f : [M] \times [N]^{k-1} \to \{0,1\}$ is $\Omega(k^{-1} \cdot N^{(k-1)/2})$ (see details in Section 8), thus our construction is optimal (up to a factor of k). Previously, this result was only known for $k = 2$ [19] (for the case that $M = N$). For $k > 2$, in the best previously known linear CDS protocol the size of the messages was $O(N^k)$ [20].

Following [9, 12, 10], we also consider functions with few inputs that return 1. We consider k-input functions f such that $|f^{-1}(1)| \leq N^\gamma$ for some $0 < \gamma < (k+1)/2$ and construct a linear CDS protocol for them with message size $O(k^3 \cdot N^{\gamma(k-1)/(k+1)} \cdot \log N)$. The same result holds for functions such that $|f^{-1}(0)| \leq N^\gamma$. These results generalize the result of [11] that constructed a CDS protocol for 2-input functions f such that $|f^{-1}(1)| \leq N^\gamma$ for some constant $1 \leq \gamma < 2$ with message size $\tilde{O}(N^{\gamma/4})$. The results of [10] imply a lower bound of $\Omega(k^{-1} \cdot N^{\gamma(k-1)/2k})$ for the message size of linear CDS protocols for k-input functions. We do not know if our construction for k-input functions with few inputs that return 1 is optimal.

As discussed above, CDS protocols imply secret-sharing schemes for uniform access structures. Thus, our results imply the existence of linear secret-sharing schemes for uniform access structures as we next elaborate. Using a family of perfect hash functions and our CDS protocols, we show that every k-uniform access structure with n parties can be realized by a linear secret-sharing scheme with share size $O(k \cdot e^k \cdot \log n \cdot \lceil n/k \rceil^{(k-1)/2})$ (a similar transformation was presented in [3]; our transformation is more efficient). Furthermore, using a transformation of [14], every k-uniform access structure with n parties can be realized by a linear secret-sharing scheme with share size $O(n \cdot 2^{n/2})$; this protocol is more efficient when $k > 0.257n$. Finally, our results imply that every k-uniform access structure with n^γ minimal authorized sets of size k can be realized by a linear secret-sharing scheme in which the size of the share of each party is $O(k^4 \cdot e^k \cdot \log^2 n \cdot \lceil n/k \rceil^{\gamma(k-1)/(k+1)})$.

1.2 Our Technique

We use the following paradigm to design multiparty linear CDS protocols; this paradigm was implicitly used to design multiparty CDS protocols [27] and PSM protocols [14]. We start with a CDS protocol for a constant number of parties and use it to construct a CDS protocol for an arbitrary number of parties.

We demonstrate this idea by describing a linear k-party CDS protocol for a k-input function $f : [N]^k \to \{0,1\}$ with complexity $O(N^{3k/4-1})$. Notice that for $k > 2$ this construction already improves the best previously known upper bound described in [20] of $O(N^k)$ for linear CDS protocols. For simplicity of the discussion, in this paragraph we only consider an even k (as explained in the technical section we also show how to handle odd values of k). Given the function f, we define a 2-input function $g : [N]^{k/2} \times [N]^{k/2} \to \{0,1\}$, where $g((x_1,\ldots,x_{k/2}),(x_{k/2+1},\ldots,x_k)) = f(x_1,\ldots,x_k)$. By [19], there is a linear 2-party CDS protocol for g with messages of size $O(N^{k/4})$. Denote the message of the first and second party in the CDS protocol for g by $m_1(x_1,\ldots,x_{k/2})$ and $m_2(x_{k/2+1},\ldots,x_k)$, respectively (these messages are also a function of the common randomness of the CDS protocol). We construct a k-party CDS protocol for f, where the first $k/2$ parties (respectively, the last $k/2$ parties) use a $k/2$-party *PSM protocol* to compute $m_1(x_1,\ldots,x_{k/2})$ (respectively, $m_2(x_{k/2+1},\ldots,x_k)$). The parties can use the PSM protocol of [18] to compute these functions; the complexity of the protocol is $O(N^{3k/4-1})$. The referee can reconstruct the messages $m_1(x_1,\ldots,x_{k/2})$ and $m_2(x_{k/2+1},\ldots,x_k)$ and use the linear reconstruction function of the CDS protocol to reconstruct the secret. The problem is that the resulting CDS protocol is not linear since the PSM protocol of [18] is not linear. However, we can use the fact that in a CDS protocol the referee knows $x_1,\ldots,x_k$ and construct a simplified version of the protocol of [18] that is linear.

We use the above approach to design a linear k-party CDS protocol with messages of size $N^{(k-1)/2}$. We first construct a new linear 3-party CDS protocol for 3-input functions; this CDS protocol generalizes the linear 2-party CDS protocol of [19]. To construct a CDS protocol for a k-input function $f : [N]^k \to \{0,1\}$ (for an odd k) we define a 3-input function $g : [N] \times [N]^{(k-1)/2} \times [N]^{(k-1)/2} \to \{0,1\}$, where $g(x_1,(x_2,\ldots,x_{(k-1)/2}),(x_{(k+1)/2},\ldots,x_k)) = f(x_1,\ldots,x_k)$; that is, we partition the parties to three sets, where the size of the first set is 1 and the sizes of the two other sets is $(k-1)/2$. We use our 3-party CDS protocol for g, and denote the messages in this protocol by m_1, m_2, m_3; in this protocol each message is of size at most $N^{(k-1)/2}$. We then show that m_2 and m_3 can be computed by efficient linear PSM protocols (where the referee knows the inputs $x_2,\ldots,x_{(k-1)/2}$ and $x_{(k+1)/2},\ldots,x_k$, respectively).

To summarize our approach, one can start with any linear CDS protocol for a small number of parties and use a linear variant of the PSM protocol of [18], in which the parties send messages enabling the referee to compute the messages of the CDS protocol. However, this transformation does not necessary result in the most efficient protocol. To construct an optimal linear k-party CDS protocol, we design a specific 3-party CDS protocol, such that its messages can be computed by efficient linear PSM protocols.

Comparison to the protocol of [27]. In a work that was done independently and in parallel to our work, Liu et al. [27] have also constructed k-party linear CDS protocols for arbitrary k-input functions with total message size $O(k \cdot N^{(k-1)/2})$ for every $k > 2$. Their protocol is somewhat different than ours, however it uses very similar ideas. We apply some optimizations in our protocol, which reduces the total message size by a factor of k compared to the protocol of [27]. Furthermore, the protocol of [27] is only described for odd values of k (using our ideas it can be transformed to a protocol for even values of k).

1.3 Related Works

Gertner et al. [20] defined CDS protocols and used them to construct symmetrically-private information retrieval protocols. They gave some constructions of CDS protocols: (1) they showed that a PSM protocol for a function implies a CDS protocol for the same function, and (2) they showed that a span program (not necessarily monotone) computing a function f implies a linear CDS protocol for f. In particular, this gives a construction from formulas and branching programs. Their result implies that for every k-input function $f : [N]^k \to \{0,1\}$ there exist a linear CDS protocol with messages of size $O(N^k)$.

Beimel et al. [13] showed that for every 2-input function $f : [N] \times [N] \to \{0,1\}$ there exists a 2-party CDS protocol in which the size of the messages is $O(N^{1/2})$. Their protocol is not linear. Gay et al. [19] constructed a linear 2-party CDS protocol for arbitrary 2-input functions with the same message size of $O(N^{1/2})$. Following the above results, Liu et al. [26] have shown that every 2-input function has a non-linear 2-party CDS protocol with messages of size $2^{O(\sqrt{\log N \log\log N})}$. To construct this CDS protocol, they reduced it to a CDS protocol for the index function and constructed a CDS protocol for the index function based on the private information retrieval protocol of Dvir and Gopi [17]. Liu et al. [27] have generalized their results to k-input functions, designing a non-linear k-party CDS protocol with messages of size $2^{O(\sqrt{k \log N} \log(k \log N))}$.

Gay et al. [19] proved lower and upper bounds on the size of the messages in linear and non-linear 2-party CDS protocols for several functions with domain of size N. For example, they proved a lower bound of $\Omega(\sqrt{\log N})$ and a matching upper bound of $O(\sqrt{\log N})$ on the messages size of linear CDS protocols for the index function and a lower bound of $\Omega(\sqrt{\log N})$ and an upper bound of $O(\log N)$ on the messages size of linear CDS protocols for the disjointness function (which returns 1 if and only if the sets represented by the inputs are disjoint) and for the inner-product function. They also proved a lower bound of $\Omega(\log\log N)$ for any CDS protocol (possibly non-linear) for these functions. Applebaum et al. [4] proved a lower bound of $\Omega(\log N)$ for any CDS protocol (possibly non-linear) for some (non-explicit) function. Applebaum et al. [5] proved a lower bound of $\log N - 3 - o(1)$ for any CDS protocol (possibly non-linear) for the inner product function. All the above lower bounds are for a one-bit secret.

Applebaum et al. [4] and Ambrona et al. [2] showed that if there is a linear 2-party CDS protocol for some function f with message size c and common random string with size r, then there is a linear CDS protocol for the complement

function $\overline{f}$ in which the message size and the common random string size is linear in c and r. Applebaum et al. [4] also showed that if there is a 2-party CDS protocol (possibly non-linear) for some function f with message size c, common random string with size r, and an error of $2^{-\kappa}$ (in the reconstruction and in the privacy), then there is a CDS protocol for $\overline{f}$ in which the message size and the common random string size are polynomial in c, r, and κ.

Another result shown in [4] is that for every 2-input function there exists a linear CDS for secrets of ℓ bits, where ℓ is exponential in N^2, in which the size of the messages is $O(\ell \cdot \log N)$. This gives an amortized message size of $O(\log N)$ per each bit of the secret, much better than the message size of $2^{O(\sqrt{\log N \log\log N})}$ shown in [26]. Applebaum and Arkis [3] improved this result and extended it to k-input functions; they showed that for every function $f : [N]^k \to \{0,1\}$ there exists a multi-linear CDS protocol for secrets of ℓ bits, where ℓ is exponential in N^k, in which the size of each of the messages sent by the parties is 4ℓ.

CDS protocols are closely related to secret-sharing schemes for uniform access structures. Basically, k-party CDS protocols for functions $f : [N]^k \to \{0,1\}$ are equivalent to secret-sharing schemes for k-partite k-uniform access structures with $k \cdot N$ parties, where a k-uniform access structure is k-partite if there is a partition of the parties to k sets $V_1, \ldots, V_k$ such that every authorized set of size k contains exactly one party from each set V_i. Two-uniform access structures, called forbidden graph access structures, where first defined by Sun and Shieh [28], and where further studied in [13, 12, 10, 3].

In particular, it was shown in [13] that there is a transformation from 2-party CDS protocols to secret-sharing schemes for 2-uniform access structures with n parties in which the share size is $O(\log n)$ times the message size in the CDS protocol; this transformation preserves linearity. Furthermore, if the size of the secret is increased, then the share size of the resulting scheme is only $O(1)$ times the message size in the CDS protocol; this transformation *does not* preserve linearity (for a linear CDS, the resulting scheme would be multi-linear). In [3], this transformation was generalized for any k, where the increase in the share size is $O(e^k \cdot \log n)$ if one wants to preserve linearity and $O(e^k)$ without preserving linearity. In this paper, we improve this transformation for short secrets, i.e., we transform k-party CDS protocols for a function *with domain of size* n/k to secret-sharing schemes for k-uniform access structures with n parties.

2 Preliminaries

2.1 Conditional Disclosure of Secrets Protocols

In this section we define k-party conditional disclosure of secrets (CDS) protocols, first presented in [20].

Definition 2.1 (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* 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 $(x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ 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 total message size of a CDS protocol $\mathcal{P}$ is the total size of the messages sent by the parties, i.e., $\sum_{i=1}^{k} \log |M_i|$.

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 only to $x_1, \ldots, x_k$, such 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 2.2 (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 $(x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 0$, every secret $s \in S$, and every k messages $(m_1, \ldots, m_k) \in M_1 \times \cdots \times M_k$,*

$$\begin{aligned} &\Pr[\text{SIM}(x_1, \ldots, x_k) = (m_1, \ldots, m_k)] \\ &= \Pr[\text{ENC}_1(x_1, s, r) = m_1, \ldots, \text{ENC}_k(x_k, s, r) = m_k], \end{aligned}$$

where the first probability is over the randomness of the simulator S and the second probability is over the choice of r from R with uniform distribution.

Informally, we say that a CDS protocol is linear if the reconstruction function of the referee is a linear function.

Definition 2.3 (Linear Conditional Disclosure of Secrets Protocols). *We say that a CDS protocol is linear over a finite field $\mathbb{F}$ if*

- *$S = \mathbb{F}$,*
- *There exists constants $\ell, \ell_1, \ldots, \ell_k$ such that $R = \mathbb{F}^\ell$ and $M_i = \mathbb{F}^{\ell_i}$ for every $i \in [k]$, and*
- *For every $x_1, \ldots, x_k \in [N]$ there exist field elements $(\alpha_{i,j_i})_{i \in [k], j_i \in [\ell_i]} \in \mathbb{F}$ such that*

$$\text{DEC}(x_1, \ldots, x_k, \text{ENC}_1(x_1, s, r), \ldots, \text{ENC}_k(x_k, s, r)) = \sum_{i \in [k], j_i \in [\ell_i]} \alpha_{i,j_i} m_{i,j_i},$$

where $\text{ENC}_i(x_i, s, r) = (m_{i,1}, \ldots, m_{i,\ell_i})$ for every $i \in [k]$.

Equivalently, we could have required that for every $i \in [k]$ and every $x_i \in X_i$ the function $\text{ENC}_i(x_i, s, r)$ is a linear function over $\mathbb{F}$ of the secret s and the field elements in $r = (r_1, \ldots, r_\ell)$ (see [24, 7] for the equivalence).

2.2 Secret-Sharing Schemes

We next present the definition of secret-sharing schemes, similar to [8, 15].

Definition 2.4 (Secret-Sharing Schemes). *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 $\min \Gamma$.*

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

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

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

CORRECTNESS. *The secret k can be reconstructed by any authorized set of parties. That is, for any set $B = \{P_{i_1}, \ldots, P_{i_{|B|}}\} \in \Gamma$, there exists a* reconstruction function $\text{Recon}_B : K_{i_1} \times \cdots \times K_{i_{|B|}} \to K$ *such that for every secret $k \in K$ and every random string $r \in R$,*

$$\text{Recon}_B \left(\Pi_B(k, r) \right) = k.$$

PRIVACY. *Every unauthorized set cannot learn anything about the secret from its shares. Formally, for any set $T \notin \Gamma$, every two secrets $a, b \in K$, and every possible vector of shares $\langle s_j \rangle_{P_j \in T}$,*

$$\Pr[\, \Pi_T(a, r) = \langle s_j \rangle_{P_j \in T} \,] = \Pr[\, \Pi_T(b, r) = \langle s_j \rangle_{P_j \in T} \,],$$

where the 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 2.5 (Linear Secret-Sharing Schemes). *Let $\Sigma = \langle \Pi, \mu \rangle$ be a secret-sharing scheme with domain of secrets K, where μ is a probability distribution on a set R and Π is a mapping from $K \times R$ to $K_1 \times K_2 \times \cdots \times K_n$. We say that Σ is a* linear secret-sharing scheme over a finite field $\mathbb{F}$ *if $K = \mathbb{F}$, the sets $R, K_1, \ldots, K_n$ are vector spaces over $\mathbb{F}$, Π is an $\mathbb{F}$-linear mapping, and μ is the uniform probability distribution over R.*

3 Linear CDS Protocols for 2 and 3 Parties

We present linear 2-party and 3-party CDS protocols. The 3-party CDS protocol will be used in Section 4 to construct k-party CDS protocols for $k > 3$. To avoid confusions, in this section we denote the parties by Alice, Bob, and Charlie.

3.1 A Linear 2-Party CDS Protocol

As a warm up, we first describe a linear 2-party CDS protocol for any 2-input function $f : [M] \times [N] \to \{0,1\}$ in which the total message size is N; i.e., the message size does not depend on M. This protocol is part of the protocol described in [19], and it is not the optimal protocol for 2 parties (in particular, by [19] there exist a linear 2-party CDS protocol for any 2-input function $f : [N] \times [N] \to \{0,1\}$ in which the message size is $O(N^{1/2})$).

In the CDS protocol, the parties, Alice and Bob, hold the inputs $x_1 \in [M]$ and $x_2 \in [N]$, respectively, and the common randomness is N uniform bits $r_1, \ldots, r_N$. We denote the secret by $s \in \{0,1\}$. Alice sends to the referee the bit

$$s \oplus \bigoplus\nolimits_{i_2 \in [N], f(x_1, i_2) = 0} r_{i_2},$$

and Bob sends the bits $r_1, \ldots, r_{x_2 - 1}, r_{x_2 + 1}, \ldots, r_N$. The message size of the protocol is $1 + (N - 1) = N$.

The correctness of the above protocol follows from the fact that if $f(x_1, x_2) = 1$, then the bit r_{x_2} is not part of the exclusive-or of the bit that Alice sends. The referee gets all the bits $r_1, \ldots, r_N$ except for the bit r_{x_2}, and in particular all the bits among $r_1, \ldots, r_N$ that are part of the exclusive-or in $s \oplus \bigoplus_{i_2 \in [N], f(x_1, i_2) = 0} r_{i_2}$. Thus, the referee can reconstruct the secret. For the privacy, we observe that if $f(x_1, x_2) = 0$, then the bit r_{x_2} is part of the exclusive-or of the bit that Alice sends, and since the referee does not get this bit from Bob, then it cannot learn any information about the secret. Formally, a simulator independently chooses N uniform bits $s', r'_1, \ldots, r'_{N-1}$ and outputs s' as the message of Alice and $r'_1, \ldots, r'_{N-1}$ as the message of Bob.

3.2 A Linear 3-Party CDS Protocol

We adapt the above protocol and construct a linear 3-party CDS protocol $\mathcal{P}_3$ for any 3-input function $f : [M] \times [N] \times [N] \to \{0,1\}$ with message size $O(N)$ (again, the message size is independent of M).

Protocol $\mathcal{P}_3$

The secret: A bit $s \in \{0,1\}$.
Inputs: Alice, Bob, and Charlie hold the inputs $x_1 \in [M]$ and $x_2, x_3 \in [N]$, respectively.
Common randomness: The three parties hold the following random bits.

- $r_1, \ldots, r_N \in \{0,1\}$.
- $q_1, \ldots, q_N \in \{0,1\}$.

The protocol:

1. Alice sends to the referee the bits $s_{i_3} = s \oplus q_{i_3} \oplus \bigoplus_{i_2 \in [N], f(x_1,i_2,i_3)=0} r_{i_2}$ for every $i_3 \in [N]$.
2. Bob sends to the referee the bits $r_1, \ldots, r_{x_2-1}, r_{x_2+1}, \ldots, r_N$.
3. Charlie sends to the referee the bit q_{x_3}.
4. If $f(x_1, x_2, x_3) = 1$, the referee computes

$$s_{x_3} \oplus q_{x_3} \oplus \bigoplus_{i_2 \in [N], f(x_1,i_2,x_3)=0} r_{i_2}.$$

Fig. 1. A linear 3-party CDS protocol $\mathcal{P}_3$ for a 3-input function $f : [M] \times [N] \times [N] \to \{0,1\}$.

Lemma 3.1. *Let $f : [M] \times [N] \times [N] \to \{0,1\}$ be a 3-input function. Then, there is a linear 3-party CDS protocol for f with total message size $O(N)$.*

Proof. The linear CDS protocol for f, denoted by $\mathcal{P}_3$, is described in Figure 1. We start with an informal description of the protocol. The parties, Alice, Bob, and Charlie, hold the inputs $x_1 \in [M]$ and $x_2, x_3 \in [N]$, respectively. The common randomness is $2N$ bits $r_1, \ldots, r_N$ and $q_1, \ldots, q_N$, and the secret is $s \in \{0,1\}$.

For every possible value $i_3 \in [N]$ of the input of Charlie, Alice sends to the referee the bit $s_{i_3} = s \oplus q_{i_3} \oplus \bigoplus_{i_2 \in [N], f(x_1,i_2,i_3)=0} r_{i_2}$ (i.e., the message that Alice sends in the 2-party CDS protocol, masked by q_{i_3}). Bob sends the bits $r_1, \ldots, r_{x_2-1}, r_{x_2+1}, \ldots, r_N$, and Charlie sends the bit q_{x_3}.

Next, we prove the correctness of $\mathcal{P}_3$. If $f(x_1, x_2, x_3) = 1$, then the bit r_{x_2} is not part of the exclusive-or in the bit s_{x_3} that Alice sends, since it contains only the bits r_{i_2} for which $f(x_1, i_2, x_3) = 0$. Thus, the referee, which gets the bit q_{x_3} and all the bits $r_1, \ldots, r_N$ except for the bit r_{x_2}, and in particular all the bits among $r_1, \ldots, r_N$ that are part of the exclusive-or in s_{x_3}, can reconstruct the secret s, as described in $\mathcal{P}_3$.

Now, we prove that $\mathcal{P}_3$ is private by constructing a simulator whose output is 3 messages, such that the distribution on the messages of $\mathcal{P}_3$ and the distribution on the messages of the simulator are the same. If $f(x_1, x_2, x_3) = 0$, then the bit r_{x_2} is part of the exclusive-or in the bit s_{x_3}, and, thus, the bit s_{x_3} is uniformly distributed given the messages of Bob and Charlie. Similarly, since the referee

does not get the bits $q_1, \ldots, q_{x_3-1}, q_{x_3+1}, \ldots, q_N$, the distribution on the bits s_{i_3}, for every $i_3 \in [N]$ such that $i_3 \neq x_3$, is uniform. Hence, the simulator independently chooses $2N$ uniform bits $s'_1, \ldots, s'_N, r'_1, \ldots, r'_{N-1}, q'$ and outputs $s'_1, \ldots, s'_N$ as the message of Alice, $r'_1, \ldots, r'_{N-1}$ as the message of Bob, and q' as the message of Charlie.

Moreover, the protocol $\mathcal{P}_3$ is linear over $\mathbb{F}_2$, since for every $x_1 \in [M]$ and $x_2, x_3 \in [N]$ the reconstruction function of the referee is a linear combination of the bits in the messages it gets. Finally, Alice sends N bits, Bob sends $N-1$ bits, and Charlie sends one bit, so the massage size of $\mathcal{P}_3$ is $N+(N-1)+1 = 2N$. □

4 Linear k-Party CDS Protocols

We use the protocol $\mathcal{P}_3$ to construct a k-party CDS protocol, for any integer k, using the approach described in the introduction. First, in Section 4.1, we show how to transform the 3-party CDS protocol $\mathcal{P}_3$ to a linear k-party CDS protocol $\mathcal{P}_k$ for any k-input function $f : [M] \times [N]^{k-1} \to \{0,1\}$, for an *odd* $k > 3$. Then, in Section 4.2, we show how we can adapt the transformation for an even $k > 3$.

4.1 A Linear k-Party CDS Protocol for an Odd k

Informal Description of the Protocol. We consider a k-input function $f : [M] \times [N]^{k-1} \to \{0,1\}$, for some odd k, and k parties $P_1, \ldots, P_k$ that hold the inputs $x_1 \in [M]$ and $x_2, \ldots, x_k \in [N]$, respectively. Let $k' = (k-1)/2$, $y_1 = x_1, y_2 = (x_2, \ldots, x_{k'+1})$, and $y_3 = (x_{k'+2}, \ldots, x_k)$, and define a 3-input function $g : [M] \times [N]^{k'} \times [N]^{k'} \to \{0,1\}$, where $g(x_1, (x_2, \ldots, x_{k'+1}), (x_{k'+2}, \ldots, x_k)) = f(x_1, \ldots, x_k)$. That is, we partition the parties into three sets, where the first set is $S_1 = \{P_1\}$, the second set is $S_2 = \{P_2, \ldots, P_{k'+1}\}$, and the third set is $S_3 = \{P_{k'+2}, \ldots, P_k\}$. Observe that $|S_2| = |S_3| = k'$.

We next describe a k-party CDS protocol $\mathcal{P}_k$ in which the parties $P_1, \ldots, P_k$ simulates the parties in the protocol $\mathcal{P}_3$ for the function g. In this simulation, party P_1 simulates Alice, the parties in S_2 simulate Bob, and the parties in S_3 simulate Charlie, as follows. We denote the simulated inputs in $\mathcal{P}_3$ by y_1, y_2, y_3 and use $h_2, h_3 \in [N]^{k'}$ as possible inputs of g in $\mathcal{P}_3$.

Simulating Alice. Party P_1 sends the bits $s_{h_3} = s \oplus q_{h_3} \oplus \bigoplus_{h_2 \in [N]^{k'}, f(x_1,h_2,h_3)=0} r_{h_2}$, for every $h_3 = (i_{k'+2}, \ldots, i_k) \in [N]^{k'}$ (exactly as in $\mathcal{P}_3$).

Simulating Bob. The parties in S_2 should send the bits r_{h_2}, for every $h_2 = (i_2, \ldots, i_{k'+1}) \in [N]^{k'}$, except for $r_{y_2} = r_{x_2,\ldots,x_{k'+1}}$. To do so, every party $P_j \in S_2$ sends to the referee all the random bits r_{h_2} for every $h_2 = (i_2, \ldots, i_{k'+1}) \in [N]^{k'}$ such that $i_j \neq x_j$. Observe that $h_2 \neq (x_2, \ldots, x_{k'+1})$ if and only if $i_j \neq x_j$ for at least one j. Thus, the parties in S_2 send the bits that they should send.

Simulating Charlie. The parties in S_3 should send the bit q_{y_3}. To do so, we share every random bit q_{h_3}, for every $h_3 \in [N]^{k'}$, between the parties in S_3 using a k'-out-of-k' secret-sharing scheme. That is, for every $h_3 = (i_{k'+2}, \ldots, i_k) \in [N]^{k'}$, we choose k' random bits $q_{h_3}^{k'+2}, \ldots, q_{h_3}^{k}$ and define $q_{h_3} = q_{h_3}^{k'+2} \oplus \cdots \oplus q_{h_3}^{k}$. Every party $P_j \in S_3$ sends the bits $q_{h_3}^{j}$ for every $h_3 = (i_{k'+2}, \ldots, i_k) \in [N]^{k'}$ such that $i_j = x_j$. Thus, the referee can reconstruct the bit $q_{y_3} = q_{x_{k'+2},\ldots,x_k}$, and cannot learn any information about the bits $(q_{h_3})_{h_3 \neq y_3}$.

As explained above, the referee in $\mathcal{P}_k$ can compute the messages in $\mathcal{P}_3$, and, thus, when $g(y_1, y_2, y_3) = 1$ (i.e., when $f(x_1, \ldots, x_k) = 1$), it can reconstruct the secret s. The message size of every party is at most $N^{k'} = N^{(k-1)/2}$, and the total massage size is $N^{k'} + k' \cdot N^{k'-1} \cdot (N-1) + k' \cdot N^{k'-1} = O(k \cdot N^{(k-1)/2})$.

Next, we show how to improve the total message size of the above protocol by a factor of k, by improving the simulations of Bob and Charlie by the parties in S_2 and S_3, respectively.

The improved simulation of the messages of Charlie is as follows. The common random string will contain bits $q_{i_j,\ldots,i_k}^{j}$, for every $j \in \{k'+2, \ldots, k\}$ and every $i_j, \ldots, i_k \in [N]$. First, let $q_{i_{k'+2},\ldots,i_k} = \bigoplus_{j=k'+2}^{k} q_{i_j,\ldots,i_k}^{j}$, for every $i_{k'+2}, \ldots, i_k \in [N]$. Party P_j, for every $j \in \{k'+2, \ldots, k\}$, sends the random bits $q_{x_j,i_{j+1}\ldots,i_k}^{j}$, for every $i_{j+1}, \ldots, i_k \in [N]$. The referee gets the bits $q_{x_{k'+2},\ldots,x_k}^{k'+2}, q_{x_{k'+3},\ldots,x_k}^{k'+3}, \ldots, q_{x_k}^{k}$, and thus can reconstruct $q_{x_{k'+2},\ldots,x_k}$. We will show that all other bits $q_{i_{k'+2},\ldots,i_k}$ remain random to the referee, and, thus, the privacy still holds.

The improved simulation of the messages of Bob is as follows. The common random string contains the bits $t_{i_j,\ldots,i_{k'+1}}$, for every $j \in \{3, \ldots, k'+1\}$ and every $i_j, \ldots, i_{k'+1} \in [N]$ (in addition to all previously mentioned bits). Party P_2 sends the random bits $r_{i_2,\ldots,i_{k'+1}}$, for every $i_2, i_3, \ldots, i_{k'+1} \in [N]$ such that $i_2 \neq x_2$ as before. In addition it also sends the bits $r_{x_2,i_3,\ldots,i_{k'+1}}$, for every $i_3, \ldots, i_{k'+1} \in [N]$, masked by random bits, that is, it sends $r_{x_2,i_3,\ldots,i_{k'+1}} \oplus t_{i_3,\ldots,i_{k'+1}}$, for every $i_3, \ldots, i_{k'+1} \in [N]$. Next, party P_3 sends all the bits $t_{i_3,\ldots,i_{k'+1}}$, for every $i_3, i_4, \ldots, i_{k'+1} \in [N]$ such that $i_3 \neq x_3$. Given those bits, the referee can learn all the bits $r_{i_2,i_3,\ldots,i_{k'+1}}$ for which $i_2 \neq x_2$, and all the bits $r_{i_2,i_3,\ldots,i_{k'+1}}$ for which $i_2 = x_2$ and $i_3 \neq x_3$. We continue in the same manner until we get to the party $P_{k'+1}$. That is, the party P_3 additionally sends the bits $t_{x_3,i_4\ldots,i_{k'+1}} \oplus t_{i_4,\ldots,i_{k'+1}}$, for every $i_4, \ldots, i_{k'+1} \in [N]$, and so on. Finally, party $P_{k'+1}$ sends only the bits $t_{i_{k'+1}}$, for every $i_{k'+1} \in [N]$ such that $i_{k'+1} \neq x_{k'+1}$.

The referee will learn only the bit $q_{x_{k'+2},x_{k'+3}\ldots,x_k}$ from the messages of the parties that simulate Charlie, and all the bits $r_{i_2,\ldots,i_{k'+1}}$, for every $i_2, \ldots, i_{k'+1} \in [N]$, except for $r_{x_2,\ldots,x_{k'+1}}$, from the messages of the parties that simulate Bob.

The size of the messages sent by parties $P_{k'+2}, \ldots, P_k$ is $N^{k'-1} + N^{k'-2} + \cdots + N + 1 < 2 \cdot N^{k'-1} = O(N^{(k-3)/2})$, and the size of the messages sent by parties $P_2, \ldots, P_{k'+1}$ is $N^{k'} + N^{k'-1} + \cdots + N^2 + N - 1 < 2 \cdot N^{k'} = O(N^{(k-1)/2})$.

Lemma 4.1. *Let $f : [M] \times [N]^{k-1} \to \{0,1\}$ be a k-input function, for some odd integer $k > 3$. Then, protocol $\mathcal{P}_k$, described in Figure 2, is a linear k-party CDS protocol for f with total message size $O(N^{(k-1)/2})$.*

Protocol $\mathcal{P}_k$ for an Odd k

The secret: A bit $s \in \{0,1\}$.
Inputs: $P_1, \ldots, P_k$ hold the inputs $x_1 \in [M]$ and $x_2, \ldots, x_k \in [N]$, respectively.
Common randomness: Let $k' = (k-1)/2$. The k parties hold the following random bits.

- $r_{i_2,\ldots,i_{k'+1}} \in \{0,1\}$, for every $i_2, \ldots, i_{k'+1} \in [N]$.
- $t_{i_j,\ldots,i_{k'+1}} \in \{0,1\}$, for every $j \in \{3, \ldots, k'+1\}$ and $i_j, \ldots, i_{k'+1} \in [N]$.
- $q^j_{i_j,\ldots,i_k} \in \{0,1\}$, for every $j \in \{k'+2, \ldots, k\}$ and $i_j, \ldots, i_k \in [N]$.

The protocol:

1. Define $q_{i_{k'+2},\ldots,i_k} = \bigoplus_{j=k'+2}^{k} q^j_{i_j,\ldots,i_k}$ for every $i_{k'+2}, \ldots, i_k \in [N]$.
   (* Simulation of Alice *)
2. Party P_1 sends to the referee the bits

   $$s_{i_{k'+2},\ldots,i_k} = s \oplus q_{i_{k'+2},\ldots,i_k} \oplus \bigoplus_{i_2,\ldots,i_{k'+1} \in [N], f(x_1,i_2,\ldots,i_{k'+1},i_{k'+2},\ldots,i_k)=0} r_{i_2,\ldots,i_{k'+1}}$$

   for every $i_{k'+2}, \ldots, i_k \in [N]$.
   (* Simulation of Bob *)
3. Party P_2 sends to the referee the bits $r_{i_2,\ldots,i_{k'+1}}$, for every $i_2, \ldots, i_{k'+1} \in [N]$ such that $i_2 \neq x_2$, and the bits $r_{x_2,i_3,\ldots,i_{k'+1}} \oplus t_{i_3,\ldots,i_{k'+1}}$, for every $i_3, \ldots, i_{k'+1} \in [N]$.
4. For every $j \in \{3, \ldots, k'\}$, party P_j sends to the referee the bits $t_{i_j,\ldots,i_{k'+1}}$, for every $i_j, \ldots, i_{k'+1} \in [N]$ such that $i_j \neq x_j$, and the bits $t_{x_j,i_{j+1}\ldots,i_{k'+1}} \oplus t_{i_{j+1},\ldots,i_{k'+1}}$, for every $i_{j+1}, \ldots, i_{k'+1} \in [N]$.
5. Party $P_{k'+1}$ sends to the referee the bits $t_{i_{k'+1}}$, for every $i_{k'+1} \in [N]$ such that $i_{k'+1} \neq x_{k'+1}$.
   (* Simulation of Charlie *)
6. For every $j \in \{k'+2, \ldots, k\}$, party P_j sends to the referee the bits $q^j_{x_j,i_{j+1}\ldots,i_k}$, for every $i_{j+1} \ldots, i_k \in [N]$.
   (* The Reconstruction of the Secret *)
7. If $f(x_1, \ldots, x_k) = 1$, the referee computes

   $$s_{x_{k'+2},\ldots,x_k} \oplus q_{x_{k'+2},\ldots,x_k} \oplus \bigoplus_{i_2,\ldots,i_{k'+1} \in [N], f(x_1,i_2,\ldots,i_{k'+1},x_{k'+2},\ldots,x_k)=0} r_{i_2,\ldots,i_{k'+1}}.$$

Fig. 2. A linear k-party CDS protocol $\mathcal{P}_k$ for a k-input function $f : [M] \times [N]^{k-1} \to \{0,1\}$, for an odd k.

Proof. Recall that $k' = (k-1)/2$. We prove that protocol $\mathcal{P}_k$ is a CDS protocol for f with message size as in the lemma. Let $g : [M] \times [N]^{k'} \times [N]^{k'} \to \{0,1\}$ be the 3-input function where $g(x_1, (x_2, \ldots, x_{k'+1}), (x_{k'+2}, \ldots, x_k)) = f(x_1, \ldots, x_k)$. We first prove that in protocol $\mathcal{P}_k$, the referee can compute the messages that the referee gets in the protocol $\mathcal{P}_3$ for g, and, thus, it can compute s if $g(x_1, (x_2, \ldots, x_{k'+1}), (x_{k'+2}, \ldots, x_k)) = 1$, i.e., if $f(x_1, \ldots, x_k) = 1$. We then prove that if $f(x_1, \ldots, x_k) = 0$, then the messages in $\mathcal{P}_k$ can be simulated since they are uniformly distributed regardless of s.

Correctness. First, we show that the referee gets the bit $q_{x_{k'+2},\ldots,x_k}$. Observe that the referee gets the bit $q^j_{x_j\ldots,x_k}$ from party P_j, for every $j \in \{k'+2, \ldots, k\}$. Thus, the referee can perform an exclusive-or between all these bits and reconstruct the bit $q_{x_{k'+2},\ldots,x_k} = \bigoplus_{j=k'+2}^{k} q^j_{x_j,\ldots,x_k}$.

Second, we show that the referee gets all the bits $r_{i_2,\ldots,i_{k'+1}}$, for every $i_2, \ldots, i_{k'+1} \in [N]$, except for the bit $r_{x_2,\ldots,x_{k'+1}}$. Fix some $(i_2, \ldots, i_{k'+1}) \neq (x_2, \ldots, x_{k'+1})$, and let $\ell \in \{2, \ldots, k'+1\}$ be the first index for which $i_\ell \neq x_\ell$. If $\ell = 2$, then the referee gets the bit $r_{i_2,\ldots,i_{k'+1}}$ from party P_2. Otherwise, the referee gets the bit $r_{x_2,\ldots,x_{\ell-1},i_\ell,\ldots i_{k'+1}} \oplus t_{x_3,\ldots,x_{\ell-1},i_\ell,\ldots i_{k'+1}}$ from party P_2, and for every $j \in \{3, \ldots, \ell-1\}$, it gets the bit $t_{x_j,\ldots,x_{\ell-1},i_\ell,\ldots i_{k'+1}} \oplus t_{x_{j+1},\ldots,x_{\ell-1},i_\ell,\ldots i_{k'+1}}$ from party P_j. Moreover, since $i_\ell \neq x_\ell$, the referee gets the bit $t_{i_\ell,\ldots i_{k'+1}}$ from party P_ℓ. Thus, the referee can perform an exclusive-or between all the above bits and reconstruct the bit $r_{i_2,\ldots,i_{k'+1}}$.

Using the above two facts, we prove the correctness of $\mathcal{P}_k$. The referee gets $s_{x_{k'+2},\ldots,x_k}, (r_{i_2,\ldots,i_{k'+1}})_{(i_2,\ldots,i_{k'+1}) \neq (x_2,\ldots,x_{k'+1})}$, and $q_{x_{k'+2},\ldots,x_k}$, i.e., the messages it would get in the protocol $\mathcal{P}_3$ for the function g. Hence, if $f(x_1, \ldots, x_k) = 1$, then $g(x_1, (x_2, \ldots, x_{k'+1}), (x_{k'+2}, \ldots, x_k)) = 1$ and the referee can reconstruct the secret s since it would have reconstructed it in $\mathcal{P}_3$, as described in $\mathcal{P}_k$.

Privacy. We prove that $\mathcal{P}_k$ is private by constructing a simulator. The simulator of $\mathcal{P}_k$ chooses independently uniform random bits as the messages sent by the parties. We show that the output of the simulator is distributed as the messages sent by the parties in the protocol $\mathcal{P}_k$ for $f(x_1, \ldots, x_k) = 0$, i.e., we show that in this case the messages in $\mathcal{P}_k$ are uniformly distributed.

First, the messages of parties $P_{k'+2}, \ldots, P_k$ contain random bits from the common randomness and each bit is only sent by one of the parties, thus, the messages sent by these parties are uniformly distributed. Next, the message of party $P_{k'+1}$ is uniformly distributed, since it contains the random bits $t_{i_{k'+1}}$, for every $i_{k'+1} \in [N]$ such that $i_{k'+1} \neq x_{k'+1}$. Given this message, the message of party $P_{k'}$ is uniformly distributed, since it contains the random bits $t_{i_{k'},i_{k'+1}}$, for every $i_{k'}, i_{k'+1} \in [N]$ such that $i_{k'} \neq x_{k'}$, and the bits $t_{x_{k'},i_{k'+1}} \oplus t_{i_{k'+1}}$ which contains the random bit $t_{x_{k'},i_{k'+1}}$, for every $i_{k'+1} \in [N]$. We continue in the same manner, and conclude that given the messages of parties $P_3, \ldots, P_{k'+1}$, the message of party P_2 is uniformly distributed, since it contains the random bits $r_{i_2,\ldots,i_{k'+1}}$, for every $i_2, \ldots, i_{k'+1} \in [N]$ such that $i_2 \neq x_2$, and the bits $r_{x_2,i_3,\ldots,i_{k'+1}} \oplus t_{i_3,\ldots,i_{k'+1}}$, for every $i_3, \ldots, i_{k'+1} \in [N]$. Thus, the messages

of parties $P_2, \ldots, P_{k'+1}$ are uniformly distributed. Note that the messages of $P_2, \ldots, P_{k'+1}$ and $P_{k'+2}, \ldots, P_k$ are independent.

We next argue that the message of P_1 is uniformly distributed given the messages of the other parties. We first prove that the bits $q_{i_{k'+2},\ldots,i_k}$, for every $(i_{k'+2}, \ldots, i_k) \neq (x_{k'+2}, \ldots, x_k)$, are uniformly distributed given the messages of $P_{k'+2}, \ldots, P_k$. Fix some $(i_{k'+2}, \ldots, i_k) \neq (x_{k'+2}, \ldots, x_k)$, and let $\ell \in \{k'+2, \ldots, k\}$ be the first index for which $i_\ell \neq x_\ell$, i.e., $(i_{k'+2}, \ldots, i_k) = (x_{k'+2}, \ldots, x_{\ell-1}, i_\ell, \ldots, i_k)$. Thus, the referee does not get the bit $q^\ell_{i_\ell,\ldots i_k}$ from party P_ℓ, and, thus, it cannot learn the bit $q_{i_{k'+2},\ldots,i_k}$, since $q^\ell_{i_\ell,\ldots i_k}$ is part of the exclusive-or in the bit $q_{i_{k'+2},\ldots,i_k}$. In the above argument, we used $q^\ell_{i_\ell,\ldots i_k}$ only for $q_{x_{k'+2},\ldots,x_{\ell-1},i_\ell,\ldots,i_k}$, thus, the set of bits $\{q_{i_{k'+2},\ldots,i_k}\}_{(i_{k'+2},\ldots,i_k)\neq(x_{k'+2},\ldots,x_k)}$ are uniformly distributed given the messages of $P_{k'+2}, \ldots, P_k$.

We next show that the referee does not learn the bit $r_{x_2,\ldots,x_{k'+1}}$. The referee gets the bit $r_{x_2,\ldots,x_{k'+1}} \oplus t_{x_3,\ldots,x_{k'+1}}$ from party P_2, and for every $j \in \{3, \ldots, k'\}$, it gets the bit $t_{x_j,\ldots,x_{k'+1}} \oplus t_{x_{j+1},\ldots,i_{k'+1}}$ from P_j. However, party $P_{k'+1}$ does not send to the referee the bit $t_{x_{k'+1}}$, so it cannot learn the bit $r_{x_2,\ldots,x_{k'+1}}$.

Now, we show that given the messages of parties $P_2, \ldots, P_k$, the message of party P_1 is uniformly distributed. Since $f(x_1, \ldots, x_k) = 0$, the bit $r_{x_{k'+2},\ldots,x_k}$ is part of the exclusive-or in the bit $s_{x_{k'+2},\ldots,x_k}$. As we have shown, the referee does not get $r_{x_{k'+2},\ldots,x_k}$, so the bit $s_{x_{k'+2},\ldots,x_k}$ is uniformly distributed. For every $(i_{k'+2}, \ldots, i_k) \neq (x_{k'+2}, \ldots, x_k)$, the bit $q_{i_{k'+2},\ldots,i_k}$ is part of the exclusive-or in the bit $s_{i_{k'+2},\ldots,i_k}$. As we have shown, the referee does not get $q_{i_{k'+2},\ldots,i_k}$, so the bit $s_{i_{k'+2},\ldots,i_k}$ is uniformly distributed. Thus, since for every $i_{k'+2}, \ldots, i_k \in [N]$ there is a unique random bit that is part of the exclusive-or in the bit $s_{i_{k'+2},\ldots,i_k}$ that cannot be learned by the referee, the bits $(s_{i_{k'+2},\ldots,i_k})_{i_{k'+2},\ldots,i_k\in[N]}$ are uniformly distributed and independent of each other and of the secret. Overall, the messages sent by the parties are uniformly distributed.

Message size. The size of the message of party P_1 is $N^{k'}$, the sizes of the messages of parties $P_2, \ldots, P_{k'+1}$ are $N^{k'}, N^{k'-1}, \ldots, N^2, N-1$, respectively, and the sizes of the messages of parties $P_{k'+2}, \ldots, P_k$ are $N^{k'-1}, N^{k'-2}, \ldots, N, 1$, respectively. Thus, the total message size of $\mathcal{P}_k$ is $N^{k'} + (N^{k'} + \cdots + N - 1) + (N^{k'-1} + \cdots + 1) < N^{k'} + 2 \cdot N^{k'} + 2 \cdot N^{k'-1} = O(N^{(k-1)/2})$. □

4.2 A Linear k-Party CDS Protocol for an Even k

Next, we adopt the CDS protocol $\mathcal{P}_k$ to even values of k. Given a k-input function $f : [M] \times [N]^{k-1} \to \{0, 1\}$, for an even k, and k parties $P_1, \ldots, P_k$ that hold the inputs $x_1 \in [M]$ and $x_2, \ldots, x_k \in [N]$, respectively, we define $k' = (k+2)/2$, $x_{k'} = (x^1_{k'}, x^2_{k'})$, where $x^1_{k'}, x^2_{k'} \in [N^{1/2}]$, and $y_1 = x_1, y_2 = (x_2, \ldots, x_{k'-1}, x^1_{k'})$, and $y_3 = (x^2_{k'}, x_{k'+1}, \ldots, x_k)$. As before, we partition the parties into three sets S_1, S_2, S_3, but now we split the input of party $P_{k'}$, and it will be in both sets S_2, S_3, with half of its input in each of them. That is, $S_1 = \{P_1\}$, $S_2 = \{P_2, \ldots, P_{k'}\}$, and $S_3 = \{P_{k'}, \ldots, P_k\}$. The protocol for an even k is the same as the protocol for an odd k, where $P_{k'}$ participates in the simulations of Bob and Charlie, in which it uses $x^1_{k'}$ and $x^2_{k'}$, respectively.

Protocol $\mathcal{P}_k$ for an Even k

The secret: A bit $s \in \{0,1\}$.
Inputs: $P_1, \ldots, P_k$ hold the inputs $x_1 \in [M]$ and $x_2, \ldots, x_k \in [N]$, respectively.
Common randomness: Let $k' = (k+2)/2$. The k parties hold the following random bits.

- $r_{i_2,\ldots,i_{k'-1},i^1_{k'}} \in \{0,1\}$, for every $i_2, \ldots, i_{k'-1} \in [N]$ and $i^1_{k'} \in [N^{1/2}]$.
- $t_{i_j,\ldots,i_{k'-1},i^1_{k'}} \in \{0,1\}$, for every $j \in \{3, \ldots, k'\}$, $i_j, \ldots, i_{k'-1} \in [N]$, and $i^1_{k'} \in [N^{1/2}]$.
- $q^{k'}_{i^2_{k'},i_{k'+1},\ldots,i_k} \in \{0,1\}$, for every $i^2_{k'} \in [N^{1/2}]$ and $i_{k'+1}, \ldots, i_k \in [N]$.
- $q^{j}_{i_j,\ldots,i_k} \in \{0,1\}$, for every $j \in \{k'+1, \ldots, k\}$ and $i_j, \ldots, i_k \in [N]$.

The protocol:

1. Define $q_{i^2_{k'},i_{k'+1},\ldots,i_k} = q^{k'}_{i^2_{k'},i_{k'+1},\ldots,i_k} \oplus \bigoplus_{j=k'+1}^{k} q^{j}_{i_j,\ldots,i_k}$ for every $i^2_{k'} \in [N^{1/2}]$ and $i_{k'+1}, \ldots, i_k \in [N]$.
   (* Simulation of Alice *)
2. Party P_1 sends to the referee the bits

   $$s_{i^2_{k'},i_{k'+1}\ldots,i_k} = s \oplus q_{i^2_{k'},i_{k'+1}\ldots,i_k} \oplus \bigoplus_{i_2,\ldots,i_{k'-1}\in[N],i^1_{k'}\in[N^{1/2}],f(x_1,i_2,\ldots,i_{k'-1},i^1_{k'},i^2_{k'},i_{k'+1},\ldots,i_k)=0} r_{i_2,\ldots,i_{k'-1},i^1_{k'}}$$

   for every $i^2_{k'} \in [N^{1/2}]$ and $i_{k'+1}, \ldots, i_k \in [N]$.
   (* Simulation of Bob *)
3. Party P_2 sends to the referee the bits $r_{i_2,\ldots,i_{k'-1},i^1_{k'}}$, for every $i_2, \ldots, i_{k'-1} \in [N]$ such that $i_2 \neq x_2$ and $i^1_{k'} \in [N^{1/2}]$, and the bits $r_{x_2,i_3\ldots,i_{k'-1},i^1_{k'}} \oplus t_{i_3,\ldots,i_{k'-1},i^1_{k'}}$, for every $i_3, \ldots, i_{k'-1} \in [N]$ and $i^1_{k'} \in [N^{1/2}]$.
4. For every $j \in \{3, \ldots, k'-1\}$, party P_j sends to the referee the bits $t_{i_j,\ldots,i_{k'-1},i^1_{k'}}$, for every $i_j, \ldots, i_{k'-1} \in [N]$ such that $i_j \neq x_j$ and $i^1_{k'} \in [N^{1/2}]$, and the bits $t_{x_j,i_{j+1}\ldots,i_{k'-1},i^1_{k'}} \oplus t_{i_{j+1},\ldots,i_{k'-1},i^1_{k'}}$, for every $i_{j+1}, \ldots, i_{k'-1} \in [N]$ and $i^1_{k'} \in [N^{1/2}]$.
5. Party $P_{k'}$, which holds the input $x_{k'} = (x^1_{k'}, x^2_{k'})$, where $x^1_{k'}, x^2_{k'} \in [N^{1/2}]$, sends to the referee the bits $t_{i^1_{k'}}$, for every $i^1_{k'} \in [N^{1/2}]$ such that $i^1_{k'} \neq x^1_{k'}$.
   (* Simulation of Charlie *)
6. Party $P_{k'}$ sends to the referee the bits $q^{k'}_{x^2_{k'},i_{k'+1},\ldots,i_k}$, for every $i_{k'+1}, \ldots, i_k \in [N]$.
7. For every $j \in \{k'+1, \ldots, k\}$, party P_j sends to the referee the bits $q^{j}_{x_j,i_{j+1},\ldots,i_k}$, for every $i_{j+1}, \ldots, i_k \in [N]$.
   (* The Reconstruction of the Secret *)
8. If $f(x_1, \ldots, x_k) = 1$, the referee computes

   $$s_{x^2_{k'},x_{k'+1}\ldots,x_k} \oplus q_{x^2_{k'},x_{k'+1},\ldots,x_k} \oplus \bigoplus_{i_2,\ldots,i_{k'-1}\in[N],i^1_{k'}\in[N^{1/2}],f(x_1,i_2,\ldots,i_{k'-1},i^1_{k'},x^2_{k'},x_{k'+1},\ldots,x_k)=0} r_{i_2,\ldots,i_{k'-1},i^1_{k'}}.$$

Fig. 3. A linear k-party CDS protocol $\mathcal{P}_k$ for a k-input function $f : [M] \times [N]^{k-1} \to \{0,1\}$, for an even k.

The protocol $\mathcal{P}_k$ for an even k described in Figure 3. The fact that now not all the inputs have the same size does not change the correctness and the privacy of the protocol. Moreover, the message size of protocol $\mathcal{P}_k$ for an even k is the same as in protocol $\mathcal{P}_k$ for an odd k.

The above explanation together with Lemma 4.1 implies the following result.

Theorem 4.2. *Let $f : [M] \times [N]^{k-1} \to \{0,1\}$ be a k-input function, for some integer $k > 2$. Then, there is a linear k-party CDS protocol for f with total message size $O(N^{(k-1)/2})$.*

5 Linear k-Party CDS Protocols for Unbalanced Functions

We show how to construct linear k-party CDS protocols for k-input functions with a small number of inputs that return 1 and for k-input functions with a small number of inputs that return 0. We start by constructing a k-party linear CDS protocol for k-input functions in which for every input x_k there are most d inputs $(x_1, \ldots, x_{k-1})$ such that $f(x_1, \ldots, x_{k-1}, x_k) = 1$. Next, we use this CDS protocol to construct a k-party linear CDS protocol for the desired functions.

First, let us present the following result from [10], which we are going to use in our basic construction.

Definition 5.1 (Degree of an Input). *Let $f : [M] \times [N]^{k-1} \to \{0,1\}$ be a k-input function. The degree of an input $x_k \in [N]$ is $|\left\{(x_1, \ldots, x_{k-1}) \in [M] \times [N]^{k-2} : f(x_1, \ldots, x_{k-1}, x_k) = 1\right\}|$.*

Claim 5.2 ([10]). *Let $f : [M] \times [N] \to \{0,1\}$ be a 2-input function in which the degree of every $x_2 \in [N]$ is at most $d \leq M$. Then, for a field $\mathbb{F}$ such that $|\mathbb{F}| \geq M$, there are M linear subspaces $V_1, \ldots, V_M \subseteq \mathbb{F}^{d+1}$ of dimension d and N vectors $\mathbf{z_1}, \ldots, \mathbf{z_N} \in \mathbb{F}^{d+1}$ such that for every $x_1 \in [M]$ and every $x_2 \in [N]$ it holds that $\mathbf{z_{x_2}} \in V_{x_1}$ if and only if $f(x_1, x_2) = 1$. Furthermore, for every $i \in [M]$, the basis of V_i is $\mathbf{v_1}, \ldots, \mathbf{v_d}$, where $\mathbf{v_j} = \mathbf{e_{j+1}} - i \cdot \mathbf{e_j}$ for every $j \in [d]$.*

These linear subspaces and vectors are used in [10] to construct the following linear 2-party CDS protocol for 2-input functions $f : [M] \times [N] \to \{0,1\}$ in which the degree of every $x_2 \in [N]$ is at most d. Alice and Bob, which hold the inputs $x_1 \in [M]$ and $x_2 \in [N]$, respectively, send the messages $\mathbf{v_1} \cdot \mathbf{r}, \ldots, \mathbf{v_d} \cdot \mathbf{r}$ and $s + \mathbf{z_{x_2}} \cdot \mathbf{r}$, respectively, where $s \in \mathbb{F}$ is the secret, $\mathbf{r} \in \mathbb{F}^{d+1}$ is the common randomness, and $\mathbf{v_1}, \ldots, \mathbf{v_d}$ are a basis of the linear subspace V_{x_1}. If $f(x_1, x_2) = 1$, then $\mathbf{z_{x_2}} \in V_{x_1}$ and there exist constants $u_1, \ldots, u_d$ such that $u_1 \cdot \mathbf{v_1} + \cdots + u_d \cdot \mathbf{v_d} = \mathbf{z_{x_2}}$. Thus, the referee can compute $u_1 \cdot \mathbf{v_1} \cdot \mathbf{r} + \cdots + u_d \cdot \mathbf{v_d} \cdot \mathbf{r} = \mathbf{z_{x_2}} \cdot \mathbf{r}$ and unmask the secret s from the message $s + \mathbf{z_{x_2}} \cdot \mathbf{r}$. Otherwise, if $f(x_1, x_2) = 0$, it can be shown, given the messages of Alice, that the distribution on $\mathbf{z_{x_2}} \cdot \mathbf{r}$ is uniform, and, thus, the referee cannot reconstruct the secret. The total message size of this CDS protocol is $(d+1) \log |\mathbb{F}|$ and the size of the secret is $\log |\mathbb{F}|$.

We show how to use these ideas to construct a linear k-party CDS protocol for k-input functions $f : [N]^k \to \{0,1\}$ in which the degree of every input

$x_k \in [N]$ of the last party is at most d, in which the message size of each party is $O(d \cdot k \cdot \log N)$. This result is non-trivial since we do not have any bound on the degree of the inputs of the first $k-1$ parties.

In the following protocol we simulate the above 2-party CDS protocol for the 2-input function $g : [N]^{k-1} \times [N] \to \{0,1\}$, where $g((x_1,\ldots,x_{k-1}),x_k) = f(x_1,\ldots,x_k)$. The first $k-1$ parties simulate Alice and the kth party simulates Bob. For this simulation, we use properties of the basis of V_i as described in Claim 5.2. The protocol in [10] does not need to use these properties.

Lemma 5.3. *Let $f : [N]^k \to \{0,1\}$ be a k-input function in which the degree of every $x_k \in [N]$ is at most $d \leq N^{k-1}$. Then, there is a linear k-party CDS protocol for f in which the message size of each of the first $k-1$ parties is $O(d \cdot k \cdot \log N)$ and the message size of the last party is $O(k \cdot \log N)$.*

Proof. Let $\mathbb{F}$ be the smallest finite field with a prime number of elements such that $|\mathbb{F}| \geq N^{k-1}$, and define $g : [N]^{k-1} \times [N] \to \{0,1\}$ as the 2-input function $g((x_1,\ldots,x_{k-1}),x_k) = f(x_1,\ldots,x_k)$, as above. Next, let $V_{1,\ldots,1},\ldots,V_{N,\ldots,N} \subseteq \mathbb{F}^{d+1}$ and $\mathbf{z_1},\ldots,\mathbf{z_N} \in \mathbb{F}^{d+1}$ be the N^{k-1} subspaces of dimension d and N vectors guarantied by Claim 5.2 for the function g. We represent the inputs of $P_1,\ldots,P_{k-1}$ as an element in $\{0,\ldots,N^{k-1}-1\}$, i.e., $(x_1,\ldots,x_{k-1}) = (x_1-1)N^{k-2} + (x_2-1)N^{k-3} + \cdots + (x_{k-2}-1)N + x_{k-1} - 1 \in \{0,\ldots,N^{k-1}-1\}$. Thus, the ith vector in the basis of $V_{x_1,\ldots,x_{k-1}}$ is

$$\begin{aligned}\mathbf{v_i} &= \mathbf{e_{i+1}} - (x_1,\ldots,x_{k-1}) \cdot \mathbf{e_i} \\ &= \mathbf{e_{i+1}} - (x_1-1)N^{k-2} \cdot \mathbf{e_i} - \cdots - (x_{k-2}-1)N \cdot \mathbf{e_i} - (x_{k-1}-1) \cdot \mathbf{e_i},\end{aligned}$$

that is, $\mathbf{v_i}$ is a sum of $k-1$ vectors, where the jth vector is determined by x_j, i.e., the first vector is $\mathbf{v_{i,1}} = \mathbf{e_{i+1}} - (x_1-1)N^{k-2} \cdot \mathbf{e_i}$ and for every $j \in \{2,\ldots,k-1\}$, the jth vector is $\mathbf{v_{i,j}} = -(x_j-1)N^{k-j-1} \cdot \mathbf{e_i}$. To simulate Alice, parties $P_1,\ldots,P_{k-1}$ should send $\mathbf{v_i} \cdot \mathbf{r}$ for every $i \in [d]$. Since $\mathbf{v_i} = \sum_{j=1}^{k-1} \mathbf{v_{i,j}}$, where P_j knows $\mathbf{v_{i,j}}$, party P_j can send $\mathbf{v_{i,j}} \cdot \mathbf{r}$. However, this discloses additional information to the referee, so we need to mask the messages of the parties. Specifically, for every $j \in \{1,\ldots,k-1\}$, the message of party P_j is $\mathbf{v_{1,j}} \cdot \mathbf{r} + r_1^j,\ldots,\mathbf{v_{d,j}} \cdot \mathbf{r} + r_d^j$, and the message of party P_k is $s + \mathbf{z_{x_k}} \cdot \mathbf{r}$, where $s \in \mathbb{F}$ is the secret and the common randomness is $\mathbf{r} \in \mathbb{F}^{d+1}$ and $r_i^j \in \mathbb{F}$, for every $j \in [k-1]$ and $i \in [d]$, such that $r_i^1 + \cdots + r_i^{k-1} = 0$ for every $i \in [d]$.

First, we prove the correctness of the protocol. If $f(x_1,\ldots,x_k) = 1$, then for every $i \in [d]$, the referee can compute $\mathbf{v_i} \cdot \mathbf{r} = \mathbf{v_{i,1}} \cdot \mathbf{r} + r_i^1 + \mathbf{v_{i,2}} \cdot \mathbf{r} + r_i^2 + \cdots + \mathbf{v_{i,k-1}} \cdot \mathbf{r} + r_i^{k-1}$ from the messages it gets. Next, since $\mathbf{z_{x_k}} \in V_{x_1,\ldots,x_{k-1}}$, there exist constants $u_1,\ldots,u_d$ such that $u_1 \cdot \mathbf{v_1} + \cdots + u_d \cdot \mathbf{v_d} = \mathbf{z_{x_k}}$. Thus, the referee can compute $u_1 \cdot \mathbf{v_1} \cdot \mathbf{r} + \cdots + u_d \cdot \mathbf{v_d} \cdot \mathbf{r} = \mathbf{z_{x_k}} \cdot \mathbf{r}$ and unmask the secret s from the message $s + \mathbf{z_{x_k}} \cdot \mathbf{r}$.

Now, we prove that the protocol is private, by constructing a simulator. The simulator independently chooses uniform random elements from $\mathbb{F}$ as the messages sent by the parties. We show that the messages sent by the parties in the protocol are uniformly distributed. Since the vectors $\mathbf{v_1},\ldots,\mathbf{v_d}$ are independent,

$\mathbf{v_1}\cdot\mathbf{r},\ldots,\mathbf{v_d}\cdot\mathbf{r}$ are uniformly distributed. By [10], given the values $\mathbf{v_1}\cdot\mathbf{r},\ldots,\mathbf{v_d}\cdot\mathbf{r}$, the message of party P_k is uniformly distributed when $g((x_1,\ldots,x_{k-1}),x_k)=0$ (i.e., when $f(x_1,\ldots,x_k)=0$). Furthermore, each of the messages of parties $P_1,\ldots,P_{k-1}$ contains d field elements, where the sum of the ith element from each of these messages is $\mathbf{v_i}\cdot\mathbf{r}$. Since we mask the messages, the messages of $P_1,\ldots,P_{k-2}$ are uniformly distributed, and the message of P_{k-1} is the random vector $(\mathbf{v_1}\cdot\mathbf{r},\ldots,\mathbf{v_d}\cdot\mathbf{r})$ minus the messages of $P_1,\ldots,P_{k-2}$, that is, the message of P_{k-1} is uniformly distributed as well.

The protocol is linear, since the reconstruction function of the referee is a linear combination of the messages it gets. The total message size of the protocol is $(k-1)\cdot O(d\cdot k\cdot\log N)+O(k\cdot\log N)=O(k^2\cdot d\cdot\log N)$. □

Next, we show how to transform a k-party CDS protocol for such functions to a k-party CDS protocol for k-input functions with a small number of inputs that return 1. The transformation in Lemma 5.4 is general and can start from any k-party CDS protocol for functions where the degree of every $x_k\in[N]$ is bounded. Moreover, if we start with a linear k-party CDS protocol, then the resulting k-party CDS protocol is also linear.

Lemma 5.4. *Let $f:[N]^k\to\{0,1\}$ be a k-input function, in which there are at most N^γ inputs $(x_1,\ldots,x_k)\in[N]^k$ such that $f(x_1,\ldots,x_k)=1$, for some $0<\gamma<k$, and assume that for every k-input function $f':[N]^k\to\{0,1\}$ such that the degree of every $x_k\in[N]$ is at most $d\leq N^{k-1}$ there is a k-party CDS protocol for f' with total message size c. Then, there is a k-party CDS protocol for f with total message size $k\cdot c+O((N^\gamma/d)^{(k-1)/2})$.*

Proof. Let S_i be the set of all the inputs $x_i\in[N]$ such that there are at most d inputs $(x_1,\ldots x_{i-1},x_{i+1},\ldots,x_k)\in[N]^{k-1}$ for which $f(x_1,\ldots,x_k)=1$, for every $i\in[k]$. By our assumption, there is a CDS protocol with message size c for the restriction of f to the domain $[N]^{i-1}\times S_i\times[N]^{k-i}$, for every $i\in[k]$ (by reordering the parties, we can apply the assumption for every $i\in[k]$).

Next, the set $[N]\setminus S_i$ contains all the inputs $x_i\in[N]$ such that there are more than d inputs $(x_1,\ldots x_{i-1},x_{i+1},\ldots,x_k)\in[N]^{k-1}$ for which $f(x_1,\ldots,x_k)=1$, and, thus, the number of inputs that return 1 of f is at least $|[N]\setminus S_i|\cdot d$. Therefore, $|[N]\setminus S_i|\leq N^\gamma/d$ for every $i\in[k]$. We use the protocol $\mathcal{P}_k$ of Theorem 4.2 to obtain a linear k-party CDS protocol with message size $O((N^\gamma/d)^{(k-1)/2})$ for the restriction of f to the domain $([N]\setminus S_1)\times([N]\setminus S_2)\times\cdots\times([N]\setminus S_k)$.

If $f(x_1,\ldots,x_k)=1$, and $x_i\in S_i$ for at least one $i\in[k]$, then the referee can reconstruct the secret from the messages it gets from the CDS protocol for the restriction of f to the corresponding domain. If $x_i\in[N]\setminus S_i$ for every $i\in[k]$, then the referee can reconstruct the secret from the messages it gets from the CDS protocol of Theorem 4.2. Otherwise, if $f(x_1,\ldots,x_k)=0$, then the referee cannot learn any information on the secret, which follows by the privacy of each of the independent CDS protocols we used.

Finally, if the CDS protocol with message size c we assume is linear, then the resulting protocol is linear, since in that case it is consist of independent linear protocols. The message size of the protocol is $k\cdot c+O((N^\gamma/d)^{(k-1)/2})$. □

We use the above transformation and our basic linear k-party CDS protocol for inputs with bounded degree to construct a linear k-party CDS protocol for k-input functions with a small number of inputs that return 1.

Theorem 5.5. *Let $f : [N]^k \to \{0,1\}$ be a k-input function in which there are at most N^γ inputs $(x_1, \ldots, x_k) \in [N]^k$ such that $f(x_1, \ldots, x_k) = 1$, for some $0 < \gamma < (k+1)/2$. Then, there is a linear k-party CDS protocol for f with total message size $O(k^3 \cdot N^{\gamma(k-1)/(k+1)} \cdot \log N)$.*

Proof. By Lemma 5.3, for every k-input function $f' : [N]^k \to \{0,1\}$ such that the degree of every $x_k \in [N]$ is at most $d \leq N^{k-1}$, there is a linear k-party CDS protocol for f' with total message size $O(k^2 \cdot d \cdot \log N)$. Thus, by Lemma 5.4, there is a linear k-party CDS protocol for f with total message size $O(k^3 \cdot d \cdot \log N + (N^\gamma/d)^{(k-1)/2})$. To minimize this expression, we require that $d = (N^\gamma/d)^{(k-1)/2}$, that is, $d = N^{\gamma(k-1)/(k+1)}$, and obtain a linear k-party CDS protocol with message size $O(k^3 \cdot d \cdot \log N) = O(k^3 \cdot N^{\gamma(k-1)/(k+1)} \cdot \log N)$. □

By a small modification in the first protocol as in [10], the same results hold also for k-input functions with s small number of inputs that return 0.

Lemma 5.6. *Let $f : [N]^k \to \{0,1\}$ be a k-input function in which the degree of every $x_k \in [N]$ is at least $N^{k-1} - d$, for some $d \leq N^{k-1}$. Then, there is a linear k-party CDS protocol for f in which the message size of each of the first $k-1$ parties is $O(d \cdot k \cdot \log N)$ and the message size of the last party is $O(k \cdot \log N)$.*

Theorem 5.7. *Let $f : [N]^k \to \{0,1\}$ be a k-input function in which there are at most N^γ inputs $(x_1, \ldots, x_k) \in [N]^k$ such that $f(x_1, \ldots, x_k) = 0$, for some $0 < \gamma < (k+1)/2$. Then, there is a linear k-party CDS protocol for f with total message size $O(k^3 \cdot N^{\gamma(k-1)/(k+1)} \cdot \log N)$.*

Note that the above results are not implied by the closure of CDS protocols to complement [4, 2] since the randomness in the protocols of Lemma 5.3 and Theorem 5.5 is too big.

6 Linear k-Party CDS Protocols for Functions with Inputs of Different Sizes

We use the protocol $\mathcal{P}_k$ to construct linear k-party CDS protocols for k-input functions with inputs of different sizes; as in $\mathcal{P}_k$, the message size in these protocols is independent of the largest input size. In the following three protocols, we assume, by reordering the parties, that $\alpha_1 \geq \alpha_i$ for every $i \in \{2, \ldots, k\}$.

Theorem 6.1. *Let $f : [N^{\alpha_1}] \times [N^{\alpha_2}] \times \cdots \times [N^{\alpha_k}] \to \{0,1\}$ be a k-input function, for some integer $k > 2$ and real numbers $\alpha_1, \ldots, \alpha_k > 0$. Then, there is a linear k-party CDS protocol for f with total message size $O(2^{k/2} \cdot N^{\sum_{i=2}^{k} \alpha_i/2})$.*

Proof. We view f as a k'-input function $f' : [N^{\alpha_1}] \times \{0,1\}^{k'-1} \to \{0,1\}$, where $k' = 1 + \sum_{i=2}^{k} \lceil \alpha_i \log N \rceil \leq k + \log N \cdot \sum_{i=2}^{k} \alpha_i$, and

$$f'(x_1, x_{2,1}, \ldots, x_{2,\lceil \alpha_2 \log N \rceil}, \ldots, x_{k,1}, \ldots, x_{k,\lceil \alpha_k \log N \rceil}) \\ = f(x_1, (x_{2,1}, \ldots, x_{2,\lceil \alpha_2 \log N \rceil}), \ldots, (x_{k,1}, \ldots, x_{k,\lceil \alpha_k \log N \rceil})).$$

We execute the linear k'-party CDS protocol $\mathcal{P}_{k'}$ promised by Theorem 4.2 for the k'-input function f', where party P_1 simulates the first party, party P_2 simulates the next $\lceil \alpha_2 \log N \rceil$ parties in the k'-party CDS protocol for f', party P_3 simulates the next $\lceil \alpha_3 \log N \rceil$ parties, and so on. Overall, since the message size of the protocol is independent of the size of the input of the first party, we get a linear k-party CDS protocol for the k-input function f with total message size $O(2^{(k'-1)/2}) = O(2^{(k+\log N \cdot \sum_{i=2}^{k} \alpha_i)/2}) = O(2^{k/2} \cdot N^{\sum_{i=2}^{k} \alpha_i/2})$. □

We present alternative linear CDS protocols for k-input functions $f : [N^{\alpha_1}] \times [N^{\alpha_2}] \times \cdots \times [N^{\alpha_k}] \to \{0,1\}$, where for some parameters we remove the factor of $2^{k/2}$ of the above protocol. We start with a linear k-party CDS protocol for such k-input functions, for an odd k.

Theorem 6.2. *Let $f : [N^{\alpha_1}] \times [N^{\alpha_2}] \times \cdots \times [N^{\alpha_k}] \to \{0,1\}$ be a k-input function, for some odd integer $k > 2$ and real numbers $\alpha_1, \ldots, \alpha_k > 0$. Then, there is a linear k-party CDS protocol for f with total message size $O(\min_{S \subset \{2,\ldots,k\}, |S|=(k-1)/2} \{N^{\sum_{i \in S} \alpha_i} + N^{\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i}\})$.*

Proof. Fix any set $S \subset \{2, \ldots, k\}$ such that $|S| = (k-1)/2$ and define $S_1 = \{P_j : j \in S\}$. By renaming the parties, we assume that $S_1 = \{P_2, \ldots, P_{(k+1)/2}\}$. We execute the linear k-party CDS protocol of Lemma 4.1 with the function f. Recall that in $\mathcal{P}_k$ party P_1 simulates Alice, the parties in $\{P_2, \ldots, P_{(k+1)/2}\}$ simulate Bob with an input from a domain of size $N^{\sum_{i \in S} \alpha_i}$, and the parties in $\{P_2, \ldots, P_k\} \setminus S_1 = \{P_{(k+3)/2}, \ldots, P_k\}$ simulate Charlie with an input from a domain of size $N^{\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i}$. The message size of party P_1 is $N^{\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i}$, the message size of parties $P_2, \ldots, P_{(k+1)/2}$ is less than $2 \cdot N^{\sum_{i \in S} \alpha_i}$, and the message size of parties $P_{(k+3)/2}, \ldots, P_k$ is less than $2 \cdot N^{\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i}$. Thus, the total message size of the protocol is $O(N^{\sum_{i \in S} \alpha_i} + N^{\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i})$. Since we can choose any set $S \subset \{2, \ldots, k\}$ of size $(k-1)/2$, the theorem follows. □

In the above CDS protocol, either $\sum_{i \in S} \alpha_i$ or $\sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i$ is at least $\sum_{i=2}^{k} \alpha_i/2$. So, the total message size in the CDS protocol of Theorem 6.2 can be reduced by a factor of at most $2^{k/2}$ compared to the CDS protocol of Theorem 6.1 (for example, when $\sum_{i \in S} \alpha_i = \sum_{i \in \{2,\ldots,k\} \setminus S} \alpha_i = \sum_{i=2}^{k} \alpha_i/2$). However, there are cases for which the total message size of the CDS protocol of Theorem 6.1 will be smaller than the total message size of the CDS protocol of Theorem 6.2 (for example, when $\alpha_1, \alpha_2 \gg \sum_{i=3}^{k} \alpha_i$).

Similarly to Theorem 6.2, we can construct a linear k-party CDS protocol for k-input functions, for an even k. As this CDS protocol is similar to the previous CDS protocol, we omit its details.

Theorem 6.3. *Let $f : [N^{\alpha_1}] \times [N^{\alpha_2}] \times \cdots \times [N^{\alpha_k}] \to \{0,1\}$ be a k-input function, for some even integer $k > 2$ and real numbers $\alpha_1, \ldots, \alpha_k > 0$. Then, there is a linear k-party CDS protocol for f with total message size $O(\min_{j\in\{2,\ldots,k\}, S\subset\{2,\ldots,k\}\setminus\{j\}, |S|=(k-2)/2}\{N^{\alpha_j/2+\sum_{i\in S}\alpha_i} + N^{\alpha_j/2+\sum_{i\in\{2,\ldots,k\}\setminus(S\cup\{j\})}\alpha_i}\})$.*

7 Linear Secret-Sharing Schemes Realizing k-Uniform Access Structures

7.1 General k-Uniform Access Structures

Recall that an access structure is k-uniform if all sets of size less than k are unauthorized, all sets of size greater than k are authorized, and the access structure specifies which sets of size k are authorized. A k-uniform access structure is k-partite if the parties can be partitioned into k sets $V_1, ..., V_k$ such that each authorized set of size k contains exactly one party from each set V_i. Basically, k-party CDS protocols are equivalent to secret-sharing schemes realizing k-partite k-uniform access structures, see, e.g., [3, Lemma 4.2]. Furthermore, this equivalence preserves linearity. Thus, our results imply the following theorem.

Corollary 7.1. *Let Γ be a k-partite k-uniform access structure with partition $V_1, ..., V_k$, where $|V_i| = N$ for every $i \in [k]$. Then, there is a linear secret-sharing scheme realizing Γ in which the share size of every party is $O(N^{(k-1)/2})$.*

We next describe a secret-sharing scheme realizing k-uniform access structure (not necessarily k-partite). To obtain this result, we use a generic transformation from secret-sharing schemes realizing k-partite k-uniform access structures to secret-sharing schemes realizing k-uniform access structure (not necessarily k-partite). This transformation is similar to the transformation in [3], however, for short secrets our transformation is more efficient. The transformation uses a family of perfect hash functions.

Definition 7.2. *A set of functions $H = \{h_i : [n] \to [k] : i \in [\ell]\}$ 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.*

It is known that if we sample $\ell = O(k \cdot e^k \cdot \log n)$ random functions $h_i : [n] \to [k]$, then we get a family of perfect hash functions with high probability. In our transformation we need that the outputs of every h_i are evenly distributed. We next supply a simple proof that such a family of perfect hash functions exists.

Claim 7.3. *There exists a family of perfect hash functions $H = \{h_i : [n] \to [k] : i \in [\ell]\}$, where $\ell = O(k \cdot e^k \cdot \log n)$, such that for every $i \in [\ell]$ and every $b \in [k]$ it holds that*

$$|\{a \in [n] : h_i(a) = b\}| \leq \lceil n/k \rceil. \tag{1}$$

Proof. We prove the existence of H using the probabilistic method. We can assume that n/k is an integer (otherwise we add dummy elements to the domain). We choose ℓ functions h_i independently, where in each stage we choose a function satisfying (1) with uniform distribution.

First, we fix a set $A \in [n]$ of size k, and choose one function h satisfying (1) with uniform distribution. We give a lower bound on the probability that $|h(A)| = k$. We can view the choice of such a function h as the following process: Choose a random permutation $\pi : [n] \to [n]$ and define $h(a) = b$ if $(b-1) \cdot n/k + 1 \leq \pi(a) \leq b \cdot n/k$ (e.g., all elements such that $\pi(a) \leq n/k$ are mapped to 1). Let $B = \pi(A) = \{\pi(a) : a \in A\}$. As π is a permutation chosen with uniform distribution, the set B is a uniformly distributed set of size k. Thus, the probability that $|h(A)| = k$ is the probability that a uniformly distributed set B of size k contains exactly one element from $(b-1) \cdot n/k + 1, (b-1) \cdot n/k + 2, \ldots, b \cdot n/k$, for every $b \in [k]$. The probability of the latter event is

$$\frac{(n/k)^k}{\binom{n}{k}} \geq \frac{(n/k)^k}{(e \cdot n/k)^k} = e^{-k}.$$

We choose $\ell = e^k \cdot (1 + k \cdot \ln n)$ functions $h_1, \ldots, h_\ell$ satisfying (1) independently with uniform distribution. Thus, the probability that every h_i is not one-to-one on a fixed A is at most $(1-e^{-k})^{e^k \cdot (1+k \cdot \ln n)} \leq e^{-(1+k \cdot \ln n)} = 1/(e \cdot n^k) < 1/(e \cdot \binom{n}{k})$. By the union bound, the probability that there exits a set A of size k such that every h_i is not one-to-one on A is less than $1/e$. This implies that there exists a family of perfect hash functions H of size $\ell = O(k \cdot e^k \cdot \log n)$ such that all functions in H satisfy (1). □

Next, we show how to transform a secret-sharing scheme realizing k-partite k-uniform access structures to a secret-sharing scheme realizing general k-uniform access structures. Moreover, if we start with a linear scheme, then the resulting scheme is also linear.

Lemma 7.4. *Let Γ be a k-uniform access structure with n parties. Assume that for every k-partite k-uniform access structure Γ' with partition $V_1, ..., V_k$, where $|V_i| \leq N$ for every $i \in [k]$, there is a secret-sharing scheme realizing Γ' in which the share size of every party is $c(k, N)$. Then, there is a secret-sharing scheme realizing Γ in which the share size of every party is $O(k \cdot e^k \cdot \log n \cdot c(k, \lceil n/k \rceil))$.*

Proof. Given a partition $\mathcal{V} = (V_1, ..., V_k)$ of the parties in Γ, we define the k-partite k-uniform access structure $\Gamma_{\mathcal{V}} \subset \Gamma$, where a set $A \in \Gamma$ is authorized in $\Gamma_{\mathcal{V}}$ if either $|A| > k$ or A contains exactly one party from each set V_i.

We use ℓ partitions $\mathcal{V}^1, \ldots, \mathcal{V}^\ell$ of the parties such that $\Gamma = \cup_{i=1}^{\ell} \Gamma_{\mathcal{V}^i}$ and realize each $\Gamma_{\mathcal{V}^i}$ independently. On one hand, every set $A \in \Gamma$ is authorized in at least one $\Gamma_{\mathcal{V}^i}$ so the parties in A can reconstruct the secret. On the other hand, every set $A \notin \Gamma$ is unauthorized in every $\Gamma_{\mathcal{V}^i}$ so the parties in A get no information on the secret. The share size of each party in the resulting scheme is ℓ times the size of the shares needed to realize $\Gamma_{\mathcal{V}^i}$.

We construct the ℓ partitions using the family of perfect hash functions $H = \{h_i : [n] \to [k] : i \in [\ell]\}$, for $\ell = O(k \cdot e^k \cdot \log n)$, guaranteed by Claim 7.3, where $\mathcal{V}^i = (h_i^{-1}(1), \ldots, h_i^{-1}(k))$. Using this family of perfect hash functions, every set in each partition is of size at most $\lceil n/k \rceil$. Moreover, by our assumption, there is a scheme realizing $\Gamma_{\mathcal{V}^i}$ in which the share size of every party is $c(k, \lceil n/k \rceil)$. This results in a scheme with share size $O(k \cdot e^k \cdot \log n \cdot c(k, \lceil n/k \rceil))$. □

The above transformation combined with Corollary 7.1 immediately gives the following result.

Theorem 7.5. *Let Γ be a k-uniform access structure with n parties. Then, there is a linear secret-sharing scheme realizing Γ in which the share size of every party is $O(k \cdot e^k \cdot \log n \cdot \lceil n/k \rceil^{(k-1)/2})$.*

When $k > 0.257n$, the above scheme is less efficient than trivial scheme with share size 2^n. We can use a transformation of [14] showing that if every n-input function $f : \{0,1\}^n \to \{0,1\}$ has a CDS protocol with messages of size c, then any k-uniform access structure with n parties has a secret-sharing scheme with share size $O(c \cdot n)$. This transformation preserves linearity. Thus, our linear CDS protocol implies a linear secret-sharing scheme realizing every k-uniform access structure, in which the share size is independent of k.

Theorem 7.6. *Let Γ be a k-uniform access structure with n parties. Then, there is a linear secret-sharing scheme realizing Γ in which the share size of every party is $O(n \cdot 2^{n/2})$.*

7.2 Sparse and Dense k-Uniform Access Structures

By the equivalence between CDS and uniform access structures, we obtain results for sparse and dense k-partite k-uniform access structures, which follows from Theorem 5.5 and Theorem 5.7.

Corollary 7.7. *Let Γ be a k-partite k-uniform access structure with partition $V_1, \ldots, V_k$, where $|V_i| = N$ for every $i \in [k]$. If $|\{A \in \Gamma : |A| = k\}| \leq N^\gamma$ or $|\{A \in \Gamma : |A| = k\}| \geq N^k - N^\gamma$, for some $0 < \gamma < (k+1)/2$, then there is a linear secret-sharing scheme realizing Γ in which the share size of every party is $O(k^3 \cdot N^{\gamma(k-1)/(k+1)} \cdot \log N)$.*

Using the transformation in Lemma 7.4, we can generalize the above result to every sparse and dense k-uniform access structure (not necessarily k-partite).

Corollary 7.8. *Let Γ be a k-uniform access structure with n parties. If $|\{A \in \Gamma : |A| = k\}| \leq n^\gamma$ or $|\{A \in \Gamma : |A| = k\}| \geq \binom{n}{k} - n^\gamma$, for some $0 < \gamma < (k+1)/2$, then there is a linear secret-sharing scheme realizing Γ in which the share size of every party is $O(k^4 \cdot e^k \cdot \log^2 n \cdot \lceil n/k \rceil^{\gamma(k-1)/(k+1)})$.*

The above results should be compared to the trivial linear scheme realizing sparse k-uniform access structures with n parties, in which we share the secret independently for every minimal authorized set of size k; in this scheme the share size of every party is $O(n^\gamma)$.

8 Lower Bounds for Linear Schemes Realizing k-Uniform Access Structures

In this section, we use results of [10] to prove lower bounds on the size of the shares in linear secret-sharing schemes realizing k-uniform access structures and on the size of the messages in linear k-party CDS protocols.

8.1 Lower Bounds on the Size of One Share and Implications to CDS Protocols

First, we show lower bounds on the share size of at least one party in every linear secret-sharing scheme realizing general k-partite k-uniform access structures.

Before we start, we need some notations and a lemma from [10]. 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. Furthermore, we say that $\rho_q(\Gamma) \leq s$ if there exists a linear secret-sharing scheme over $\mathbb{F}_q$ realizing Γ such that each share in the scheme contains at most s field elements.

Lemma 8.1 ([10]). *For every prime power q and integers s, r, n such that $s > \log n$, the number of access structures Γ with n parties, rank r, and $\rho_q(\Gamma) \leq s$ is at most $2^{2rns^2 \cdot \log q}$.*

Theorem 8.2. *For most k-partite k-uniform access structures Γ with partition $V_1, \ldots, V_k$, where $|V_i| = N$ for every $i \in [k]$, the share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-1} \cdot N^{(k-1)/2})$.*

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

We next bound the number of k-partite k-uniform access structures Γ that can be realized by a secret-sharing scheme in which the share size of every party is at most θ. Recall that in k-uniform access structures all sets of size $k+1$ are authorized, that is, its rank is at most $k+1$.

By Lemma 8.1, the number of k-partite k-uniform access structures Γ with $k \cdot N$ parties and $\rho_q(\Gamma) \leq \theta / \log q$, is at most $2^{2(k+1) \cdot kN \cdot (\theta / \log q)^2 \cdot \log q} < 2^{2(k+1) \cdot kN \cdot \theta^2}$. Since we are counting linear schemes, we need to sum the number of the access structures that realized by linear schemes for every possible finite field (there are at most $2^{k^{-1} \cdot N^{(k-1)/2}}$ such fields, because $q \leq 2^{k^{-1} \cdot N^{(k-1)/2}}$). Consider the access structures that realized by linear schemes in which the size of the share of every party is $\theta < k^{-1} \cdot N^{(k-1)/2}$. The number of such access structures is at most $2^{k^{-1} \cdot N^{(k-1)/2}} \cdot 2^{2(k+1) \cdot kN \cdot \theta^2} = 2^{k^{-1} \cdot N^{(k-1)/2} + 2(k+1) \cdot kN \cdot \theta^2}$.

On the other hand, the number of k-partite k-uniform access structures Γ, where the size of every part is N, is 2^{N^k}. Thus, if half of the k-partite

k-uniform access structures Γ, where the size of every part is N, have linear secret-sharing schemes in which the share size of every party is at most θ, then $2^{k^{-1}\cdot N^{(k-1)/2}+2(k+1)\cdot kN\cdot\theta^2} \geq \frac{1}{2}\cdot 2^{N^k}$, i.e., $k^{-1}\cdot N^{(k-1)/2}+2(k+1)\cdot kN\cdot\theta^2 \geq N^k-1$, so $\theta = \Omega(k^{-1}\cdot N^{(k-1)/2})$. □

By [3, Lemma 4.2], we get the following corollary for k-party CDS protocols.

Corollary 8.3. *For most k-input functions $f : [N]^k \to \{0,1\}$, the message size of at least one party in every linear k-party CDS protocol for f is $\Omega(k^{-1}\cdot N^{(k-1)/2})$.*

As we show in Theorem 4.2, this bound is tight up to a factor of k.

Sparse and Dense k-Uniform Access Structures.

Theorem 8.4. *Let $0 \leq \gamma \leq k$ be some real number. There exists a k-partite k-uniform access structure Γ with partition $V_1,\ldots,V_k$, where $|V_i| = N$ for every $i \in [k]$ and $|\{A \in \Gamma : |A| = k\}| \leq N^\gamma$, such that the share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-1}\cdot N^{\gamma(k-1)/2k})$. Furthermore, there exists a k-partite k-uniform access structure Γ with partition $V_1,\ldots,V_k$, where $|V_i| = N$ for every $i \in [k]$ and $|\{A \in \Gamma : |A| = k\}| \geq N^k - N^\gamma$, such that the share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-1}\cdot N^{\gamma(k-1)/2k})$.*

Proof. By Theorem 8.2, for every N there exists a k-partite k-uniform access structure Γ_N with N parties in every part such that the share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing the access structure Γ_N is $\Omega(k^{-1}\cdot N^{(k-1)/2})$. We use this k-partite k-uniform access structure (with fewer parties) to construct a sparse k-partite k-uniform access structure Γ with N parties in every part. Let $V_1,\ldots,V_k$ be disjoint sets of parties of size N. For every $i \in [k]$, we fix an arbitrary set of parties $V_i' \subset V_i$ of size $N' = N^{\gamma/k}$, and construct the k-partite k-uniform access structure $\Gamma_{N'}$ with parties $V_1' \cup \cdots \cup V_k'$. We define Γ as the access structure with parties $V_1 \cup \cdots \cup V_k$ that contains all sets in $\Gamma_{N'}$ and all sets of size at least $k+1$.

Since all minimal authorized sets of size k in Γ contain exactly one party from each V_i' (for every $i \in [k]$), the number of minimal authorized sets of size k is at most $(N')^k = (N^{\gamma/k})^k = N^\gamma$. The share size of at least one party for sharing a one-bit secret in every linear secret-sharing scheme realizing $\Gamma_{N'}$ (and, hence, Γ) is $\Omega(k^{-1}\cdot(N^{\gamma/k})^{(k-1)/2}) = \Omega(k^{-1}\cdot N^{\gamma(k-1)/2k}) = \Omega(k^{-1}\cdot N^{\gamma/2-\gamma/2k})$.

To construct a dense k-partite k-uniform access structure with at least $N^k - N^\gamma$ minimal sets of size k that requires large shares in every linear scheme realizing it, we use a similar construction, however, we add all sets of size k with exactly k parties from different parts that contain at least one party in $V_i \setminus V_i'$ for some $i \in [k]$. Similar analysis implies that the resulting k-partite k-uniform access structure has at least $N^k - N^\gamma$ minimal authorized sets of size k and the share size of at least one party for sharing a one-bit secret in every linear scheme realizing this k-partite k-uniform access structure is $\Omega(k^{-1}\cdot N^{\gamma(k-1)/2k})$. □

Again, by [3, Lemma 4.2], we get the following results.

Corollary 8.5. *Let $0 \leq \gamma \leq k$ be some real number. There exists a k-input function $f : [N]^k \to \{0,1\}$ such that $|\{(x_1,\ldots,x_k) : f(x_1,\ldots,x_k) = 1\}| \leq N^\gamma$, in which the message size of at least one party in every linear k-party CDS protocol for f is $\Omega(k^{-1} \cdot N^{\gamma(k-1)/2k})$. Furthermore, there exists a k-input function $f : [N]^k \to \{0,1\}$ such that $|\{(x_1,\ldots,x_k) : f(x_1,\ldots,x_k) = 0\}| \leq N^\gamma$, in which the message size of at least one party in every linear k-party CDS protocol for f is $\Omega(k^{-1} \cdot N^{\gamma(k-1)/2k})$.*

8.2 Lower Bounds on the Total Share Size

Next, we show lower bounds on the *total* share size in every linear secret-sharing scheme realizing k-uniform access structures.

Theorem 8.6. *For most k-uniform access structures Γ with n parties, the total share size for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-(k+3)/2} \cdot n^{(k+1)/2})$.*

Proof. If we share a one-bit secret using a linear secret-sharing scheme over $\mathbb{F}_q$ with shares containing S field elements, then the total share size is $S \cdot \log q$. For the total share size to be less than $k^{-(k+3)/2} \cdot n^{(k+1)/2}$, it must be that $q \leq 2^{k^{-(k+3)/2} \cdot n^{(k-1)/2}}$ (otherwise, each share contains more than $k^{-(k+3)/2} \cdot n^{(k-1)/2}$ bits, and the total share size will be more than $k^{-(k+3)/2} \cdot n^{(k+1)/2}$), and, furthermore, $S \cdot \log q \leq k^{-(k+3)/2} \cdot n^{(k+1)/2}$.

Denote the parties in Γ by P. First, we count the number of linear schemes realizing k-uniform access structures Γ over $\mathbb{F}_q$ with shares containing S field elements. Let B be the set of size at most n/k containing all the parties such that the share of each one of them containing more than $k \cdot S/n$ field elements. The set $P \setminus B$ contains all the parties such that the share of each one of them containing at most $k \cdot S/n$ field elements. We can add parties to B such that $|B| = n/k$, and the share of every party in $P \setminus B$ is still containing at most $k \cdot S/n$ field elements.

By Lemma 8.1, the number of k-uniform access structures over $\mathbb{F}_q$ with parties in $P \setminus B$ such that there exists linear schemes realizing them in which the share of every party containing at most $k \cdot S/n$ field elements is $2^{2(k+1) \cdot n(1-1/k) \cdot (kS/n)^2 \cdot \log q}$.

The number of sets with k parties that intersect B is the number of sets with k parties in P minus the number of sets with k parties contained in $P \setminus B$, i.e., $\binom{n}{k} - \binom{n(k-1)/k}{k} > (1-(1-1/k)^k)\binom{n}{k}$. Moreover, the number of possible choices of the set B is $\binom{n}{n/k}$.

Thus, the number of k-uniform access structures Γ over $\mathbb{F}_q$ with linear schemes realizing them in which the shares containing S field elements is $\binom{n}{n/k} \cdot 2^{(1-(1-1/k)^k)\binom{n}{k}} \cdot 2^{2(k+1) \cdot n(1-1/k) \cdot (kS/n)^2 \cdot \log q} = \exp\left(O\left((1-(1-1/k)^k)\binom{n}{k} + \frac{k^3 \cdot S^2 \cdot \log q}{n}\right)\right)$.

Since we are counting linear schemes, we need to sum the number of the access structures that realized by linear schemes for every possible finite field (there are at most $2^{k^{-(k+3)/2} \cdot n^{(k-1)/2}}$ such fields, because $q \leq 2^{k^{-(k+3)/2} \cdot n^{(k-1)/2}}$). Consider the access structures that realized by linear schemes with total share size at most $S \cdot \log q = \Theta < k^{-(k+3)/2} \cdot n^{(k+1)/2}$ (so here $S = \Theta / \log q$). The number of such schemes is at most $\exp\left(O\left(k^{-(k+3)/2} \cdot n^{(k-1)/2} + (1-(1-1/k)^k)\binom{n}{k} + \frac{k^3 \cdot \Theta^2}{n}\right)\right)$.

Additionally, the number of k-uniform access structures Γ with n parties is $2^{\binom{n}{k}}$. 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 $\exp\left(O\left(k^{-(k+3)/2} \cdot n^{(k-1)/2} + (1-(1-1/k)^k)\binom{n}{k} + \frac{k^3 \cdot \Theta^2}{n}\right)\right) \geq \exp\left(\binom{n}{k} - 1\right)$, i.e., $\exp\left(O\left(k^{-(k+3)/2} \cdot n^{(k-1)/2} + \frac{k^3 \cdot \Theta^2}{n}\right)\right) \geq \exp\left(\Omega\left((1-1/k)^k \binom{n}{k}\right)\right) \geq \exp\left(\Omega\left(\frac{n^k}{k^k}\right)\right)$, so we get that $\Theta = \Omega(k^{-(k+3)/2} \cdot n^{(k+1)/2})$. □

As we show in Theorem 7.5, for a constant k this bound is tight up to a logarithmic factor.

Sparse and Dense k-Uniform Access Structures.

Theorem 8.7. *Let $1 \leq \gamma \leq k$ be some real number. There exists a k-uniform access structure Γ with n parties and $|\{A \in \Gamma : |A| = k\}| \leq n^\gamma$, such that the total share size for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-(k+3)/2} \cdot n^{(\gamma+1)/2})$. Furthermore, there exists a k-uniform access structure Γ with n parties and $|\{A \in \Gamma : |A| = k\}| \geq \binom{n}{k} - n^\gamma$, such that the total share size for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ is $\Omega(k^{-(k+3)/2} \cdot n^{(\gamma+1)/2})$.*

Proof. By Theorem 8.6, for every n there exists a k-uniform access structure with n parties such that the total share size for sharing a one-bit secret in every linear secret-sharing scheme realizing it is $\Omega(k^{-(k+3)/2} \cdot n^{(k+1)/2})$. Denote the parties in Γ by P. We use this k-uniform access structure (with fewer parties) to construct a sparse k-uniform access structure Γ with n parties. We partition the parties of P to $n' = n^{(k-\gamma)/(k-1)}$ disjoint sets of parties $V_1, \ldots, V_{n'}$, where $|V_i| = n/n' = n^{(\gamma-1)/(k-1)}$ for every $i \in [n']$. We construct a copy of a k-uniform access structure from Theorem 8.6 with $n/n' = n^{(\gamma-1)/(k-1)}$ parties among the parties of V_i, and denote this k-uniform access structure by Γ_i, for every $i \in [n']$. There are no authorized sets contain parties from different sets from $V_1, \ldots, V_{n'}$.

Since every authorized set in this construction contains parties from the same set V_i (for some $i \in [n']$), the number of authorized sets is at most $n' \cdot \binom{n/n'}{k} \leq n' \cdot (n/n')^k = n^{(k-\gamma)/(k-1)} \cdot (n^{(\gamma-1)/(k-1)})^k = n^{(k-\gamma+k\gamma-k)/(k-1)} = n^{\gamma(k-1)/(k-1)} = n^\gamma$. The total share size for sharing a one-bit secret in every linear secret-sharing scheme realizing Γ_i (for every $i \in [n']$) is $\Omega(k^{-(k+3)/2} \cdot (n^{(\gamma-1)/(k-1)})^{(k+1)/2}) = \Omega(k^{-(k+3)/2} \cdot n^{(\gamma-1)(k+1)/(2(k-1))}) = \Omega(k^{-(k+3)/2} \cdot n^{(k\gamma+\gamma-k-1)/(2(k-1))})$. Thus, the total share size for sharing a one-bit secret in every linear secret-sharing

scheme realizing Γ is $n' \cdot \Omega(k^{-(k+3)/2} \cdot n^{(k\gamma+\gamma-k-1)/(2(k-1))}) = \Omega(k^{-(k+3)/2} \cdot n^{(k-\gamma)/(k-1)+(k\gamma+\gamma-k-1)/(2(k-1))}) = \Omega(k^{-(k+3)/2} \cdot n^{(k\gamma-\gamma+k-1)/(2(k-1))}) = \Omega(k^{-(k+3)/2} \cdot n^{(\gamma+1)(k-1)/(2(k-1))}) = \Omega(k^{-(k+3)/2} \cdot n^{(\gamma+1)/2})$.

To construct a dense k-uniform access structures with at least $\binom{n}{k} - n^\gamma$ authorizes sets that requires large shares in every linear scheme realizing it, we use a similar construction, however, we add all sets with exactly k parties, in which not all the paries are in the same set V_i, for some $i \in [n']$. Similar analysis implies that the resulting k-uniform access structure has at least $\binom{n}{k} - n^\gamma$ authorizes sets and the total share size for sharing a one-bit secret in every linear scheme realizing this k-uniform access structure is $\Omega(k^{-(k+3)/2} \cdot n^{(\gamma+1)/2})$. □

References

1. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 118–134 (2001)
2. Ambrona, M., Barthe, G., Schmidt, B.: Generic transformations of predicate encodings: Constructions and applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 36–66. Springer-Verlag (2017)
3. Applebaum, B., Arkis, B.: Conditional disclosure of secrets and d-uniform secret sharing with constant information rate. Tech. rep., Electronic Colloquium on Computational Complexity, www.eccc.uni-trier.de/eccc/ (2017), to appear in TCC 2018.
4. Applebaum, B., Arkis, B., Raykov, P., Vasudevan, P.N.: Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 727–757 (2017)
5. Applebaum, B., Holenstein, T., Mishra, M., Shayevitz, O.: The communication complexity of private simultaneous messages, revisited. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. pp. 261–286. LNCS, Springer-Verlag (2018)
6. Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577 (2014)
7. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D. thesis, Technion (1996), www.cs.bgu.ac.il/ beimel/pub.html
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.: Secret-sharing schemes for very dense graphs. J. of Cryptology 29(2), 336–362 (2016)
10. Beimel, A., Farràs, O., Mintz, Y., Peter, N.: Linear secret-sharing schemes for forbidden graph access structures. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017. LNCS, vol. 10678, pp. 394–423. Springer-Verlag (2017)
11. Beimel, A., Farràs, O., Mintz, Y., Peter, N.: Linear secret-sharing schemes for forbidden graph access structures. Tech. Rep. 2017/940, IACR Cryptology ePrint Archive (2017)
12. Beimel, A., Farràs, O., Peter, N.: Secret sharing schemes for dense forbidden graphs. In: Zikas, V., Prisco, R.D. (eds.) SCN 2016. pp. 509–528 (2016)
13. Beimel, A., Ishai, Y., Kumaresan, R., Kushilevitz, E.: On the cryptographic complexity of the worst functions. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 317–342. Springer-Verlag (2014)

14. Beimel, A., Kushilevitz, E., Nissim, P.: The complexity of multiparty PSM protocols and related models. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. pp. 287–318. LNCS, Springer-Verlag (2018)
15. Chor, B., Kushilevitz, E.: Secret sharing over infinite domains. J. of Cryptology 6(2), 87–96 (1993)
16. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334 (2000)
17. Dvir, Z., Gopi, S.: 2-server PIR with sub-polynomial communication. In: 47th STOC 2015. pp. 577–584 (2015)
18. Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation. In: 26th STOC 1994. pp. 554–563 (1994)
19. Gay, R., Kerenidis, I., Wee, H.: Communication complexity of conditional disclosure of secrets and attribute-based encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 485–502. Springer-Verlag (2015)
20. 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)
21. Ishai, Y., Kushilevitz, E.: Private simultaneous messages protocols with applications. In: 5th Israel Symp. on Theory of Computing and Systems. pp. 174–183 (1997)
22. Ishai, Y., Wee, H.: Partial garbling schemes and their applications. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) 41st ICALP. vol. 8572, pp. 650–662. Springer (2014)
23. 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).
24. Karchmer, M., Wigderson, A.: On span programs. In: 8th Structure in Complexity Theory. pp. 102–111 (1993)
25. Liu, T., Vaikuntanathan, V.: Breaking the circuit-size barrier in secret sharing. In: 50th STOC 2018. pp. 699–708 (2018)
26. Liu, T., Vaikuntanathan, V., Wee, H.: Conditional disclosure of secrets via non-linear reconstruction. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 758–790. Springer-Verlag (2017)
27. Liu, T., Vaikuntanathan, V., Wee, H.: Towards breaking the exponential barrier for general secret sharing. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. pp. 567–596. LNCS, Springer-Verlag (2018)
28. Sun, H., Shieh, S.: Secret sharing in graph-based prohibited structures. In: INFOCOM '97. pp. 718–724 (1997)
29. Wee, H.: Dual system encryption via predicate encodings. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 616–637. Springer-Verlag (2014)